April 2008

Kite Power for Heifer International's Overlook Farm

Gabriel Photeos Baldwin
Worcester Polytechnic Institute

Michael Joseph Sangermano
Worcester Polytechnic Institute

Nicholas Stephen Urko
Worcester Polytechnic Institute

Peter Michael Bertoli
Worcester Polytechnic Institute

Taylor A. LaLonde
Worcester Polytechnic Institute

Project # DJO-0408

Kite Power for Heifer International's Overlook Farm

An Interactive Qualifying Project Report

submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

by

____

Gabriel Baldwin

____

Peter Bertoli

____

Taylor LaLonde

____

Michael Sangermano

____

Nick Urko

April 18, 2008

____

Professor David Olinger, Advisor

Abstract

We developed a set of educational exhibits to educate the general public about a new renewable energy technology—Wind Power from Kites. In this technology, large kites are used to extract power from the wind. These exhibits will be used in conjunction with a full-scale kite power demonstrator to be permanently displayed at Overlook Farm, a Heifer International site located in Rutland, Massachusetts, which serves as a living museum dedicated to educating the general public about sustainable development. The exhibits that we created include a simple scale-model replica of the demonstrator, virtual animations, a website that details work done on kite power at Worcester Polytechnic Institute, and an electrical system that demonstrates the potential uses of kite power in the developing world. A Major Qualifying Project developed the kite power demonstrator concurrently with this project.

Table of Contents

Table of Figures

Table of Tables

1. Introduction

Every day millions of people in the developing world depend on a limited supply of electricity for their basic needs. Supplying these people even small amounts of electricity drastically improves their living standards. In Namibia, a country located in Southern Africa, it has been shown that providing even small sources of electricity increases local enterprise (Frey et al., 2007). In a stable electric grid electricity powers everything from our lights and refrigerators to our cell phones and computers. However, in rural areas of developing countries, reliable energy is not always available for basic necessities. In remote areas where extending the grid would be prohibitively expensive, alternative energy systems are one option for providing electricity. Throughout the developing world wind power as well as solar and bio-fuel power sources are used to provide basic lighting, water pumping, communications, sanitation, and other essential needs to villagers that previously had no regular access to electricity (F.D.J. Nieuwenhout et al., 2000) (Luque, Hegedus, & Knovel, 2003).

The primary focus of the wind industry within the developed world has been large scale wind turbines. Still significant focus still rests with small scale wind turbines designed for remote locations that have no access to the grid. Wind is not new as a source of power as it has been used for centuries with its beginnings in grinding grain and irrigation. These first generation windmills had many blades and were housed on buildings at the top of windy hills (Layton, 2006).

As the development of wind turbines, the primary device to capture the winds energy, has continued it has run into several problems. The most significant problem for wind turbines has been its high cost. It has been too expensive for developing nations to pursue extensively. A problem that limits wind turbines potential area of use is that it requires a relatively high minimum wind speed to produce any power at all. Wind turbines have run against several environmental regulation issues. One of these environmental issues is that wind turbines kill birds. This was a significant problem for the older turbines that the blades rotated quickly and/or were built where many birds migrate. Even though the newer turbines kill many fewer birds the stigma still exists. The second environmental issue is that some view the turbines as unsightly and destroying the natural landscape. For instance the Cape Wind project on Nantucket Sound has faced fierce resistance from inhabitance of nearby areas because it would interfere with their sea view (Seccombe, 2008). Another environmental issue is the sound pollution that the turbines produce. These problems reduce wind turbine usability in developed and developing countries.

In order to utilize the available and valuable wind resource without being hindered by the wind turbine's problems another method to generate power must be used. Analyzing several different methods of wind power capture devices and investigating and pursuing the most promising will possibly provide a device that would retains the advantages of wind turbines, but that is also low-cost, more friendly to the environment, and usable in regions with low wind speeds where wind turbines are not economical. Using large kites instead of wind turbines has the potential to give more people in the developing world access to wind power since kites are economical in lower speed Class 2 wind regions, whereas wind turbines are not. Kites have the potential to reach much higher altitudes, where the wind energy potential is much higher. Kites pacify the environmental problems of visual pollution, noise pollution, and bird kills associated with wind turbines. We estimate that the overall cost savings of the wind power from kites method will be 50% over comparably sized wind turbines. (Olinger, 2008)

Professor David Olinger at the Worcester Polytechnic Institute (WPI) has been researching the wind power from kites concept since 2006. A formal group of students completing their senior engineering project, formally known as a Major Qualifying Project (MQP), lead by Olinger worked on designing a prototype and began building one in 2006-2007. In 2007-2008 two student groups worked concurrently on solving different aspects of the project with one completing their MQP and the other completing their junior year project that combines technology with society or Interactive Qualifying Project (IQP). The MQP project group has focused on developing the rotating arm structure, kite control mechanism, kite angle of attack change mechanism, and the power conversion mechanism. This group has been building, testing, and implementing a one-kilowatt kite power prototype.

Our group, the IQP section, sought to answer the following two questions: What is the best way to educate the general public visiting Overlook Farm about the need for a low-cost wind power system in developing nations? How best describe the operation of the demonstrator and its potential impact on progress and sustainability in developing nations? To accomplish this we developed educational tools for Heifer International's Overlook Farm including a simple scale-model replica of the demonstrator, a dynamic simulation involving a virtual animation of the kite power demonstrator, a website highlighting key aspects of the project, and an electrical system that converts mechanical motion into useable electricity for later use.

1.1 Energy Crisis

We are currently in the midst of an energy crisis. This crisis consists of two parts: the main component of the energy crisis of the past, the increase in price due to lack of energy resources, and a part being recently realized, the environmental and human damage being caused by our current consumption and acquiring of fossil fuels.

In September of 2005, crude oil was being traded at under $25 per barrel on the New York Mercantile Exchange. At the end of February, 2008 the trading price hit over $100 per barrel. This increase in prices is due to a slowdown in the discovery of new supplies, as well turmoil in the Middle East. With this rapid price increase, the current cost per barrel of oil is approaching that of the inflation adjusted price in 1980, a period of economic recession ("Oil price hovers near record high", 2008). If an increase on such scales continues, it could cause significant economic problems in a number of countries as people and industries can no longer afford to finance their daily activities.

Some estimates of our current energy resources give longevity of our fossils fuels, based on current rates of discovery and increases in usage, of less than 300 years (Kruger, 2006). Other estimates, however, give much longer longevity estimates. Tester puts estimates of the longevity of coal, if used as the only energy source, at 846 years (Tester et. al., 2005). If this number is true, then we have no immediate worry of running out of energy supplies, especially considering this number does not take into account any other energy sources. However, we have a much greater and immediate issue to consider; the impact of the usage of these fossil fuels on the environment.

One of the biggest issues currently being considered is the effect of the burning of these fossil fuels on our global climate and air quality. The amount of Carbon Dioxide in the air, a very significant greenhouse gas, has risen from 280 parts per million in 1750 to over 380 parts per million today. Along with this increase, the average temperature on the Earth has risen about .6° C during the 1900s (Lauber, 2005). In the United States, the EPA monitors six major air pollutants carbon monoxide, lead, nitrogen dioxide, ozone, sulfur dioxide, and particulate matter. The creation of energy for transportation, industrial, and home usage are the greatest contributors to the release of these six air pollutants. These six air pollutants can cause of a range of conditions in humans such as decrease in circulatory transport of oxygen, increase in vulnerability to respiratory pathogens, increased asthma attacks, neurological disabilities, and respiratory diseases ("Definition of Black Lung Disease", 2004). The release of these and other hazardous air pollutants is regulated, but as long as thee fuels are being consumed for energy needs, some amount of pollutants will be released, and will have a detrimental impact on humans, plants, and animals.

The collection of fossil fuels poses a hazard to those workers who have the job of collection, and also to the environment in the area of collection of the resource. The collection of the major source of energy for the United States, Coal, provides great health risk to those involved in its collection. Approximately one in twenty coal miners are found to be suffering from black lung, a condition which leads to trouble breathing, and it is estimated to kill 1,500 past coal miner's every year ("Definition of Black Lung Disease", 2004). The coal miners also risk losing their lives in mine collapse, which is a very serious issue for miners in China. Thousands of China's five million coal miners die every year due to poor safety practices by those running the mines (Lim, 2008).

The transportation and storage of our fuels also poses great risk to not only those workers who transport it, but many humans who typically come into close contact with transportation trucks, or storage facilities like gas stations. There is a very real issue of fire or explosion if proper safety measures are not taken by those handling the fuel. A crash, a poor practice in transferring fuels, or poor regulation of storage containers can lead to leakage of very easily ignited fuels. The spilling of fuels due to poor transportation can also have a great impact on an ecosystem, and heart a great amount of both animal and plant life (Tester et. al., 2005).

In order to mitigate the effects of fossil fuels, we have to reduce our dependence on them. We need to switch our focus to energy sources that do not require the discharge of pollutants, and the endangering of human, plant, and animal life in order to acquire. We need to make a change toward the use of alternative fuels sources not only in those countries that are currently established, but also in the developing world. Alternative sources must be made cheaper, and must be established as major sources of energy in developing countries so as to avoid the addition of even more pollutants, and a painful transition later. There are a number of solutions being developed for the production of energy to solve energy problems in the world in general, and some specifically developed for areas without large electrical grid coverage. While the focus of this project is on wind power, it is appropriate to briefly review some other renewable energy sources.

One company that is interested in both producing energy and reducing carbon emissions is GreenFuel (www.greenfuelonline.com). GreenFuel's technology is made to be implemented in currently operating, or future operations of any type that give off high amounts of carbon dioxide. The waste smoke material is run through GreenFuel's system and, using the sun as its energy, algae is grown that absorbs the carbon dioxide and uses it to grow and create energy. The algae grown in the GreenFuel system can then be converted into biodiesel or ethanol for use in cars, converted for use in animal feed, or used as biomass for combustion. This system not only helps to lessen carbon dioxide emissions from energy plants, but also turns that pollutant into a useful form of energy, lessening the need for exploration, extraction, and transportation of more fuels causing more environmental and human damage.

Solix (www.solixbiofuels.com) another company specializing in biofuels from algae, points out some of the major advantages of this type of biofuel production. One major advantage of this type of biofuel production over other types of production, such as from soybeans or corn, is that algae can be grown in land that is otherwise useless for crop growing due to its not requiring soil for growth. In some areas lands that contains trees is being cleared to make way for the growing of biodiesel crops. Those trees, however, are also a major element in the reduction of free carbon dioxide. Also, in other areas, biodiesel crops are being grown instead of food crops due to the greater economic gain. This fact could cause food shortages in some developing countries. Algae growing facilities are also estimated to produce 30 to 100 times more fuel in the same amount of area as soybean production, and also require 99% less water. These facts make it not only a viable solution in developed nations, but also in developing countries so as to allow for efficient usage of otherwise useless land and to ensure that valuable food crop land is not used for fuel production.

Another company specializing in alternative energy sources is Nanosolar (www.nanosolar.com). Nanosolar has just recently began the production of a new, easier to produce, and more efficient solar panel that they say will put solar energy costs on par with that of the cost of production of electricity from coal. This is a major step, as coal is currently our cheapest source for production of energy. This technology could be used in solar farm type operations, in personal use on homes, and would be quite useful for developing nations in bringing power to areas off the grid. Solar panels could be set up for individuals with batteries as backup storage so as to provide energy even during the nighttime in off-grid areas.

Another potential source of alternative energy is biomass. The production and combustion of biomass is a carbon neutral process, because the growing of the biomass uses simply sunlight as a fuel, and uses large amounts of carbon Dioxide in its growth. The growth of most plants for biomass, however, does require a large amount of space and time for growth, and yields the equivalent of a low-grade coal. Biomass can, however, be burnt along with coal so although it may not be a reliable fuel source it can be used when acquired to try to lessen dependence on fossil fuels, biomass also does not release a number of the harmful pollutants that the burning of other sources of fuel releases. Also, waste biomass such as leftover parts of plants from food crop harvesting, or other trash can be burnt along with coal for electricity production or, in the case of developing areas, alone for heating or cooking (Tester et. al., 2005).

Windmill power is another renewable energy source that should be part of a national energy strategy. Bergey Windpower Company (www.bergey.com), provides a large range of options to customers looking to invest in personal wind power projects that give a range of sizes of installations, as well as a range of uses. They provide systems that can be battery attached for off grid power during non-peak times, systems for those connected to the grid so that energy can be credited to them on their meter to be used during non-peak power production times, and also systems specifically designed for pumping applications. However, these systems do require a minimum wind speed for production of power. For Bergey's lowest power windmill, the minimum speed for power production is 5.6 mph, and one would want a much higher average wind speed for any significant power generation. This wind speed requirement makes wind power installations very cost inefficient in many areas.

Current wind tower can only reach certain heights due to the increasing cost of higher towers. There is, however, untapped potential at these higher altitudes. Getting a kite to these higher heights, however, is far more cost effective. A longer and stronger tether is required, rather than a very massive metal structure. This project focuses on wind power, and on the concept of using kites to produce power form the wind in areas where it was not before practical, and making wind power more economical in those areas where it is already implemented.

1.2 Need for Renewable Energy in Developing Nations

Electricity helps improve living conditions in developing nations by helping to provide essential living needs along with improving general quality of life. Water is one of the most essential elements in supporting human life. An estimated 1.1 billion people yearly are not able to maintain an adequate supply of useable water (WHO, 2005), making the lack of water one of the largest challenges to world health. One of the main problems in obtaining necessary water supplies is the lack of a pure source. Many people are forced to take water from stagnate or polluted above-ground water sources. Using such tainted water sources can easily lead to the spread of disease. The implementation of electric water pumps in these areas could help alleviate this water crisis. Such pumps could provide naturally filtered water from underground sources. Another benefit of electric pumps is that they drastically cut down the time needed to obtain water. Manual extraction of water from clean wells can take over one hour per day per family (Caro, 1982). Electric wells can automatically provide the necessary water supply, effectively lifting the time burden involved with obtaining fresh water.

Electricity also helps provide cleaner, safer food. Electrically powered refrigerators are essential in keeping many food types safe from spoiling. Especially in hot climates, meats, dairy products, and fruits quickly spoil. Refrigerators help preserve the quality of the various food types, meaning that, when eaten, refrigerated food is generally safer than non refrigerated food. Electrically power stoves also help provide adequate food supplies. It does this by lifting the time burden involved in acquiring wood or other flue for a stove, removing a major fire hazard, and reducing the time required to cook food.

Electricity can improve general quality of life by providing lighting, heating, and helping to improve general hygiene. With no electricity, nighttime lighting must be provided by a lamp, open fire, or candle. These sources of light are all a fire hazard and take must be either fed frequently or replaced. Electric lights effectively eliminate the associated fire hazard and are long lasting and effortless to replace. Though electric heaters, electricity helps reduce the time necessary to collect fuel and lessen the fire danger. Lastly, electricity can improve the general level of hygiene by helping to provide clean water, electricity can aid in the cleaning of cloths, dishes, and people (Caro, 1982). These improvements all allow for more free time, allowing people to spend more time on more productive tasks or leisure activities.

2. Background Research

Rising concerns over the planet's limited resources had paved the way for the development of technologies that harnesses the energy of renewable resources. Wind power is one such resource. As altitude increases, the strength of the wind also increases, and the energy that can be produced is proportional to the cube of wind speed. Conventional wind turbines are approximately 50 meters high, but kites can easily reach altitudes in the 100 to 150 meter range. This is advantage that kites have over wind turbines, as a kite at 150 meters with the same area as a turbine at 50 meters will generate approximately twice the power (Furey, 2007).

2.1 Advantages of Kite Power vs. Turbine Power

2.1.1 Cost

It is anticipated that kite power will be more economical than wind turbines. This will grant people in developing nations cheaper access to energy, which will help them prosper. Table 1 shows the costs between a kite mechanism and a wind turbine. It has been estimated that power generation from a kite device will surpass the power generation from a turbine, once the technology is perfected.

Table 1: Capital Cost Estimation (Olinger, 2008)

KITE		WIND TURBINE[2]	
Kite(10 m^2) + Tether [1]	\$ 900	Turbine +Generator	\$ 2,590
Power conversion mechanism	\$ 800	Tower (100 ft)	2,450
Electronics, AC-DC Inverter Generator	2,000	Electronics, AC-DC Inverter	1800
Batteries (24V, 9 kW-hr) [3]	1,000	Batteries (24V, 9 kW-hr)	1,000
Accessories	500	Accessories	500
Total	**\$ 5,200**	**Total**	**\$ 8,340**

[1] Based on prices from Powerline Sports, Seabrook NH.
[2] Based on Bergey XL.1 wind turbine rated at 1000W at 20 mph, 2.5 meter diameter rotor
[3] Battery storage (no connection to grid) assumed. Batteries and accessory systems identical since both the kite and wind turbine must accommodate variable speed operation.

2.1.2 Increased Available Power at Higher Altitudes

As altitude increases, so does wind speed, and so does the amount of power that can be generated. Figure 1 shows the amount of power that can be generated from the wind at corresponding altitudes. It is important to note that small-scale wind turbines rarely exceed a height of 50 meters, because it is too expensive to build such a tall turbine. Because of this, turbines do not have access to the powerful upper altitude winds. Kites can access these high altitude winds, and therefore have a greater potential for generating energy. The different quantities of power that can be generated by turbines and kites can be seen in Table 2 (Olinger, 2008).

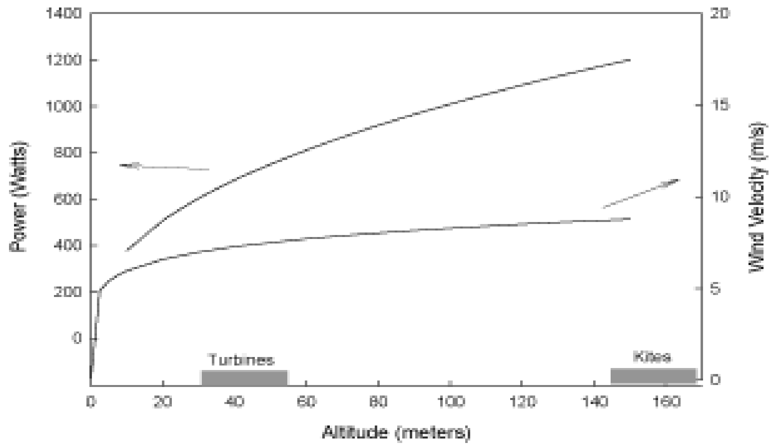

Fig. 2 Power output for turbine or kite with A = 10 m^2 area based on Wind Class 4 (see Table I). A power coefficient of C_p = 0.15 is assumed. The variation in wind velocity with altitude in Earth boundary layer is also shown. A 1/7th power law is assumed to calculate the velocity distribution. 1 m/s = 2.2 mph.

Figure 1: Generated Power (Olinger, 2008)

Table 2: Wind Power Classes (Olinger, 2008)

Wind Class	50 METERS TURBINE		150 METERS KITE	
	Power Density (W/m^2)	Wind Speed[1] m/s (mph)	Power Density (W/m^2)	Wind Speed[2] m/s (mph)
1	0-200	0 – 5.6 (0-12.5)	0-320	0 - 6.6 (0-14.6)
2	200-300	5.6-6.4 (12.5-14.3)	320-480	6.6-7.5 (14.6-16.7)
3	300-400	6.4-7.0 (14.3-15.7)	480-640	7.5-8.2 (16.7-18.4)
4	400-500	7.0-7.5 (15.7-16.8)	640-800	8.2-8.8 (18.4-19.7)
5	500-600	7.5-8.0 (16.8-17.9)	800-960	8.8-9.4 (19.7-20.9)
6	600-800	8.0-8.8 (17.9 – 19.7)	960-1280	9.4-10.3 (20.9-23.0)
7	> 800	> 8.8 (> 19.7)	>1280	>10.3 (>23.0)

[1]Mean wind speed based on Rayleigh speed distribution of equivalent mean wind power density.
[2]Vertical extrapolation of wind speed based on 1/7th power law.

2.1.3 Visual/Spatial

Kites are much more visually pleasing than wind turbines. Local opposition for a kite-powered device should be less than for a turbine. It is likely that many people will not even notice a single kite in the sky. Also, these devices take up much less ground space than turbines do, and have a much lesser impact on the local environment and wildlife (Furey, 2007).

2.1.4 Noise

There are two kinds noise pollution created by a wind turbine. The first kind of noise pollution is the aerodynamic noise created by the rotors passing through the air. A kite will be higher in the air, and will not generate a noticeable amount of noise. The second kind of noise pollution is the mechanical noise from the generator. Advances in this technology have eliminated most of this noise for wind turbines, and we believe the same technology can be applied to a kite power system ("Environmental Concerns," 2008).

2.1.5 Pollution

Wind energy is much less harmful to our environment than many of our current sources. Coal and other fossil fuel power plants generate mercury, sulfur dioxide, and nitrogen oxide emissions, which are all harmful to our environment. Nuclear power generates radioactive waste, which must be disposed of at manmade facilities. Wind energy operates cleanly, with no harmful substances being produced. Mining operations for fossil fuels and uranium destroy habitats and endanger the local wildlife. Dams and hydroelectric power endanger aquatic wildlife, either by blocking their path downstream or by heating the water to unlivable temperatures. Wind energy is a renewable resource, and does not require the destruction of the environment ("Save the Loon with Wind Energy…", 2008).

2.1.6 Avian Mortality Issue

When the technology is completed, the kites will be operating at altitudes above where birds normally fly. The control lines are a much smaller target than the blades of a wind turbine, and will result in less bird fatalities. Table 3 outlines some leading causes of bird deaths. Deaths resulting from kites are not present on this table, because there are very few recorded incidents of it.

Table 3: Avian Mortality ("Environmental Concerns", 2008)

Structure	Bird Deaths (wide range incorporates many studies)
Vehicles	60 million - 80 million
Buildings and Windows	98 million - 980 million
Power lines	174 million
Communicative Towers	4 million - 50 million
Wind Generation Facilities	10,000 - 40,000

2.2 FAA Regulations

There are some FAA regulations that will need to be observed when using this technology. In certain regions, stations may need to be properly equipped with such things as radar and radio to identify and warn any stray airplanes. Ideally, these devices would be set up in remote areas that do not have such strict airspace rules, which is likely the case in developing nations. An interesting idea thought up by the creators of KiteGen is to have such devices in areas that have already been established as no flight zones, such as the area surrounding a nuclear power plant.

2.3 Previous Kite Power Research

Dr. J. S. Goela was one of the first people to investigate the feasibility of using kites to harness wind energy. Dr. Goela and his research assistants at the Indian Institute of Technology in Kanpur published a number of yearly reports documenting his progress.

In the report published in 1983, Goela investigates a system in which the back and forth motion of a kite would be converted into up and down motion of a bucket in a deep well. He determined steady state equations of a kite in the air to determine line tensions and power output (Goela, 1983). In order to get optimal power and kite flying reliability, Goela's team proposed many different kite designs. Figure 2 shows the different kite designs:

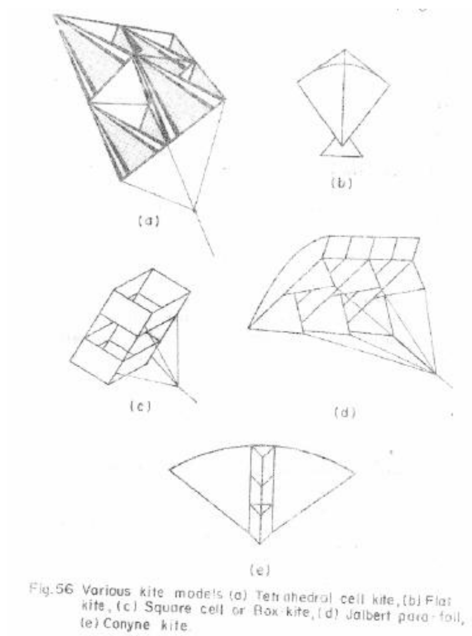

Figure 2: Goela's Kite Designs (Goela, 1983)

Each different kite design was tested in a large wind tunnel and force measurements were taken at different angles of attack. Goela and his team eventually settled on the Conyne kite because it proved to be very stable and had a good amount of lifting force. This kite is shown in Figure 3.

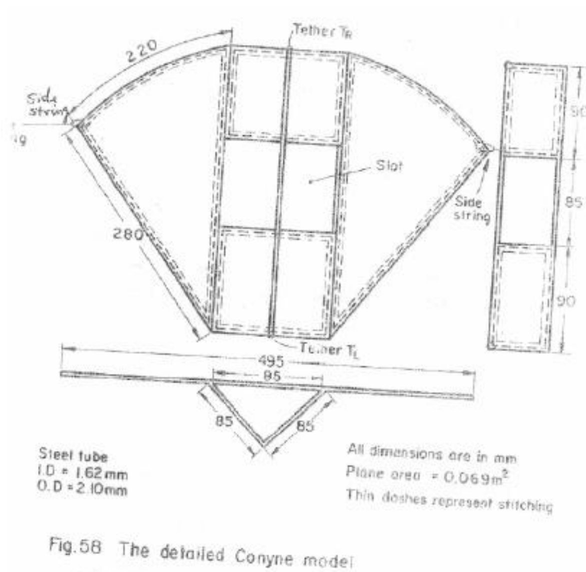

Figure 3: Goela's Final Kite Design (Goela, 1983)

The mechanism proposed by Dr. Goela's team was designed to lift a bucket from a deep well using a kite. The design incorporated a balanced beam on a fulcrum with springs attached to one end of the beam. These springs were used to change the angle of attack of the kite, causing the arm to ascend and descend (Goela, 1983). A schematic of this design is shown in Figure 4.

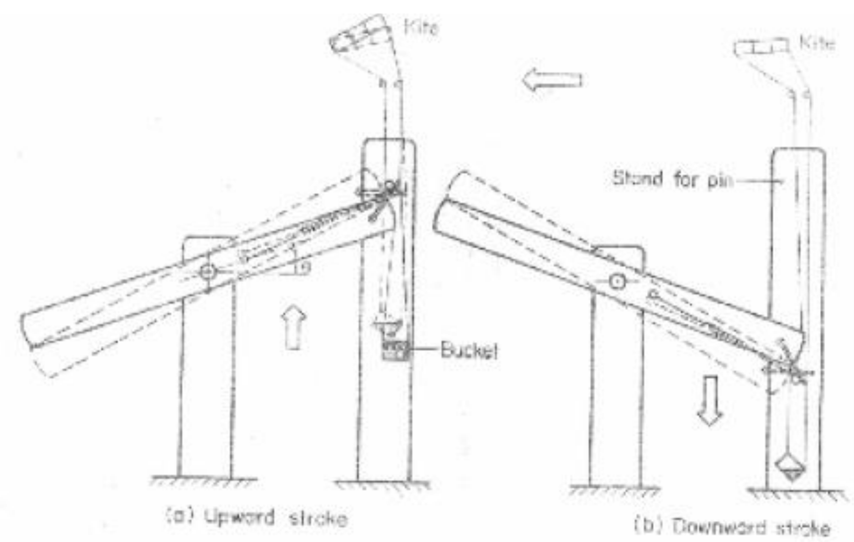

Figure 4: Goela's Kite Powered Mechanism (Goela, 1983)

Goela's research was vital to a 2006-2007 MQP project at WPI titled "Wind Power From Kites." This MQP used much of the research that Goela and his team conducted. This MQP expanded on his research and wanted to further investigate the feasibility of using kites to harness the power from wind. The MQP team investigated many different designs for a mechanism that would convert the wind energy into a usable form of energy. Eventually, a design using a kite tethered to a rocking arm was settled upon (see Figure 5). It was determined that this design had the very good angle of attack control, energy conversion, and roll control. The energy is converted using a sprag clutch system. When the arm is lifted, the chain will spin the sprag clutch, thus spinning an electrical generator (Blouin, 2007).

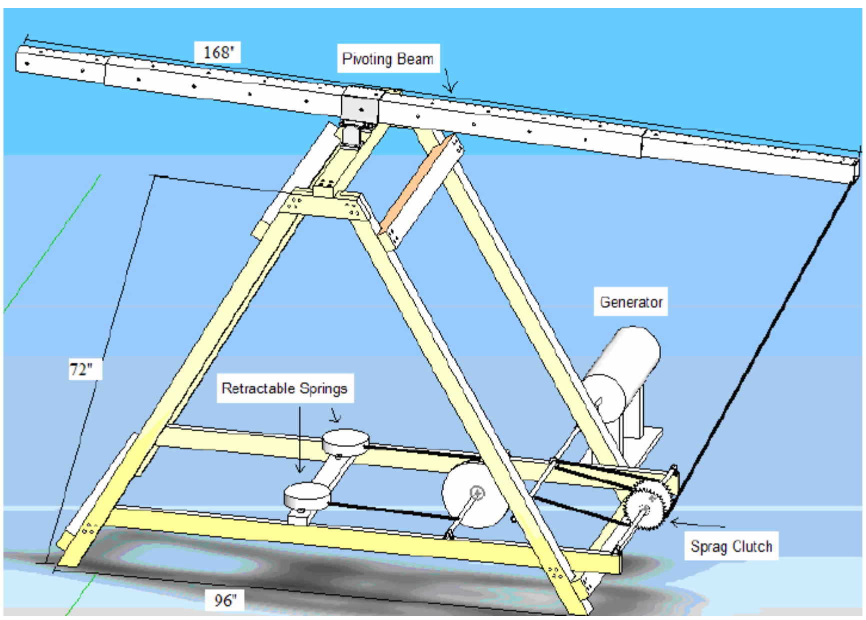

Figure 5: 2006-2007 MQP Mechanism Design (Blouin, 2007)

In order to analyze the potential power output of the mechanism, the steady state equations developed by Dr. Goela were used for simulations. There was also dynamic model made using differential equations. Dr. Goela determined equations of motion that describe the dynamics of the kite during a power cycle. Equations were obtained for the ascent and descent phases of kite motion (Goela, 2008). These simulations were used to predict the kite motion and the power generated by the mechanism. The simulations demonstrated that using kites to harness wind power is in fact feasible.

David Lang (Lang, 2005) investigates several different schemes for kite power generation and determined which designs would be the most feasible. Lang compares six different design configurations using a detailed decision matrix. Each design was rated on a number of criteria and the best were determined to be the best options. The two highest scoring designs were the “Reel” and the “KiwiGEN.” The “Reel” (shown in Figure 6) involves harnessing the energy from the kite as it is pulled out and wound back in. The energy is harnessed using a series of mechanical and electrical components. The “KiwiGEN” (shown in Figure 7) is essentially a huge merry-go-round with a number of kites attached to the outer edge.

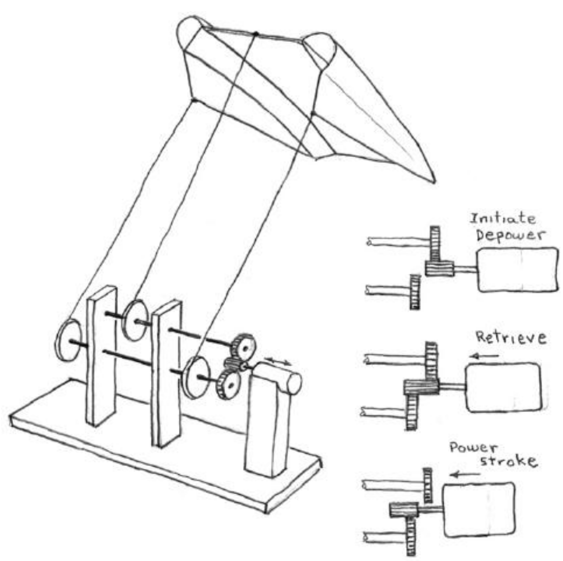

Figure 6: The Reel (Lang, 2005)

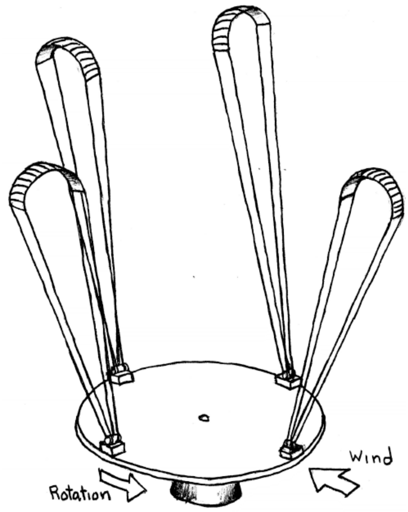

Figure 7: The KiwiGEN (Lang 2005)

Kite Gen Research is a company that is developing new technology to harness wind energy. Their goal is to create a new kind of power plant, which operates at a cheaper cost and generates equivalent power to current power installations. When completed, it is estimated that Kite Gen will produce one gigawatt of power for the cost of 1.5 euros per megawatt hour, which is significantly cheaper than the current 43 euros per megawatt hour (Martinelli, 2008). They use the high wind speed of the troposphere (lower atmosphere, 800 – 1000 meters) to power their kites, which in turn are hooked up to and rotate the central structure, generating power. The positions of the kites are monitored and controlled by very sophisticated sensors in order to optimize energy generation ("Kite Gen", 2007). Figure 8 shows a concept rendering of the system.

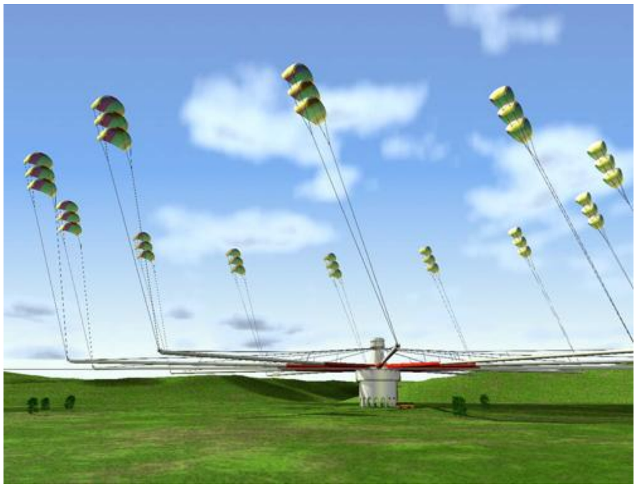

Figure 8: Kite Powered Wind Mill Schematic (Martinelli, 2008)

Another result of kite research is the Laddermill project. This device has several wings or kites that are all connected to a cable. As the wings are ascending along the cable, they are positioned for optimum lift, and as the wings are descending along the cable they are positioned for minimal drag. This cable is guided around a wheel on the ground that, when turned, generates electricity (Ockels, 2008). Figure 9 is a simple drawing of this concept.

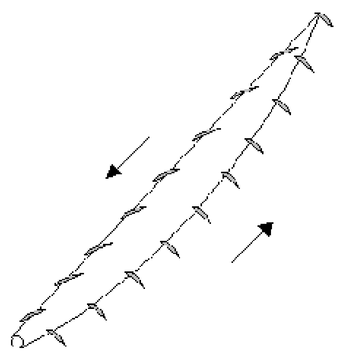

Figure 9: Electricity Generation Device Mechanism (Ockels, 2008)

Currently, there are several companies that provide and outfit sailing vessels with kites to reduce fuel costs. One such company is Kite For Sail. These kites harness the wind, and act as an auxiliary power source. These kites can be attached to any kind of vessel, and reduce fuel and operation costs. The expected savings of a standard inter-island Hawaii shipping outfitted with such a kite is tens of thousands of gallons of oil, and hundreds of tons less carbon dioxide ("Kite For Sail…", 2007). Figure 10 shows kites in use by ships for energy purposes.

Figure 10: Kite Powered Ship ("Kite For Sail…", 2007)

2.4 Previous Projects Done at Overlook Farm

Overlook Farm is located in Rutland, MA and is run by Heifer International. Heifer International focuses efforts on issues of hunger, poverty, and the environment by providing sustainable solutions to these issues. Overlook farm serves as one of Heifer International's "Learning Centers" that work toward educating the public about sustainability efforts, and how they can help to solve poverty and environmental issues worldwide. Heifer International focuses on helping people in need to get the skills they need to not just meet their immediate needs, but to ensure that they can continue to improve their lives. There have been previous MQP and IQP projects at WPI that have worked with Overlook Farm to develop educational exhibits related to sustainable development. These are reviewed next.

The Overlook Farm Learning Center allows the public to gain new understanding of international issues by providing working demonstrations of sustainable solutions through use of experiential and interactive activities. WPI has interacted with Overlook Farm in the past with several IQP projects. One such project, which was conducted in 2000, had students work with faculty at the farm to build a wind and solar powered electrical generation system for a barn and provide educational tools such as a website and onsite information services (Wong, 2000).

In a 1999-2000 WPI MQP project, a team of students developed an aquaponics, integrated aquaculture and hydroponics system, to be used at Heifer Project International (HPI) in Rutland, MA. The system consisted of a hydroponic bed (basically a tray on which plants can grow) that allows water to flow by the plants to provide hydration and nutrition. The plants help the fish by removing fish waste which is used by plants as fertilizer. Also, the plants grown "can be used as food for people or recycled as food for the fish by chopping it up and putting it back in the tank." The system, along with an educational display, was created to provide HPI an educational exhibit using an aquaponics approach to providing the world's hungry with an economical food source (Beauvais, 2000).

Our kite power IQP project maximizes the educational benefits of Environmental Protection Agency's P3 award program by extending the concepts of people, prosperity, and the planet beyond the university setting to the general public through collaboration with Overlook Farm (Olinger, 2008).

2.5 2008 Wind Power from Kites MQP

Figure 11: MQP Team with the Wind Power Full-Scale Demonstrator

Figure 11 shows the MQP team with the Wind Power Full-Scale Demonstrator. This demonstrator is an optimized design from the 2007 Wind Power from Kites Project. The 2008 project focused on generating power by attaching a kite to the rocking arm mechanism. The A-Frame is the main wooden structure and allows for a stable footing due to its large footprint, while still allowing for the rocking arm to move up and down without interruption. The kite attaches to the end of the arm through a control bar. This control bar is constructed to help keep the kite in the air and allow for powering and depowering of the kite. The control bar is attached to a sliding weight mechanism which allows for this powering and depowering cycle. The Kite Control Mechanism keeps the kite from crashing in situations where there is a strong crosswind. When the crosswind blows the kite will drift to one side and, without any control, it would eventually dive to the ground. The Kite Control Mechanism pulls on the Kite Control Bar opposite of the kite's side-to-side travel direction to try to bring it back to the center point.

The oscillating arm is attached to a rowing machine mechanism. The rowing machine mechanism allows for power to be produced in the up-stroke of the arm. A very important feature of the rowing machine is that it allows for the reeling in of the cable on the down-stroke without any effect on the rest of the system. When the cable is pulled from the rowing machine, it spins a shaft. That spinning shaft is then attached to another high speed shaft through the use of sprockets and chain. The rowing machine attached shaft has a very large sprocket, and the high speed shaft has a much smaller sprocket, allowing for it to spin much faster. The high speed shaft is then attached to a generator for the production and storing of electrical energy. The power conversion system can be seen in Figure 12.

Figure 12: Power Conversion System

3. Project Goals

This project focused on answering the following questions:

1. What is the best way to educate the general public visiting Overlook Farm about the need for a low-cost wind power system in developing nations?
2. How can we best describe the operation of the full-scale kite power demonstrator and its potential impact on progress and sustainability in developing nations?

With these questions in mind the following goals were established:

- To educate the general public visiting Overlook Farm about a new renewable energy concept – Wind Power from Kites.
- To create a scale model replica of the large scale kite power demonstrator at Overlook Farm for the purposes of demonstrating the system when the full-scale demonstrator cannot be displayed.
- To create a virtual animation of the kite power demonstrator that animates the motion of the kite, tether, rocking arm and power conversion mechanism.
- To create an electrical system for the full-scale demonstrator to convert the generator shaft rotation into usable electrical energy and store it in a battery.
- To create electrical demonstrations that display typical uses of the electricity created by the kite power system in developing nations.
- To create a website that will allow for people around the world to have access to information on the WPI Kite Power Team's work on the kite power system.

4. Methodology

4.1 Interaction with Overlook Farm

In 2006-2007 Professor Olinger had many discussions with Dale Perkins, the manager of Overlook Farm. Professor Olinger was interested in educating the public about harnessing wind power using kites. Our group gave presentations to both Dale Perkins and the Educational Director Todd Montgomery on the dates 12/03/07 and 2/25/08. During those meetings, we discussed different ways of educating the public about this new form of power generation. From these discussions and presentations, we decided to develop a scale model replica, a virtual animation, an electrical system and demonstration and a Wiki website.

4.2 Scale Model Replica

The scale model replica is a platform that can be more easily brought to presentations in order to show the workings of the full scale demonstrator. The scale model has two forms. One form is a fan powered system that goes through an oscillating up and down motion similar to that of the full scale demonstrator. The second form of the scale model is a hand-powered replica that models the sliding weight which changes the angle of attack of the kite, as well as the power train used to convert the up and down energy of the system to electrical power. Figure 13 and Figure 14 show the wind powered and hand powered replicas.

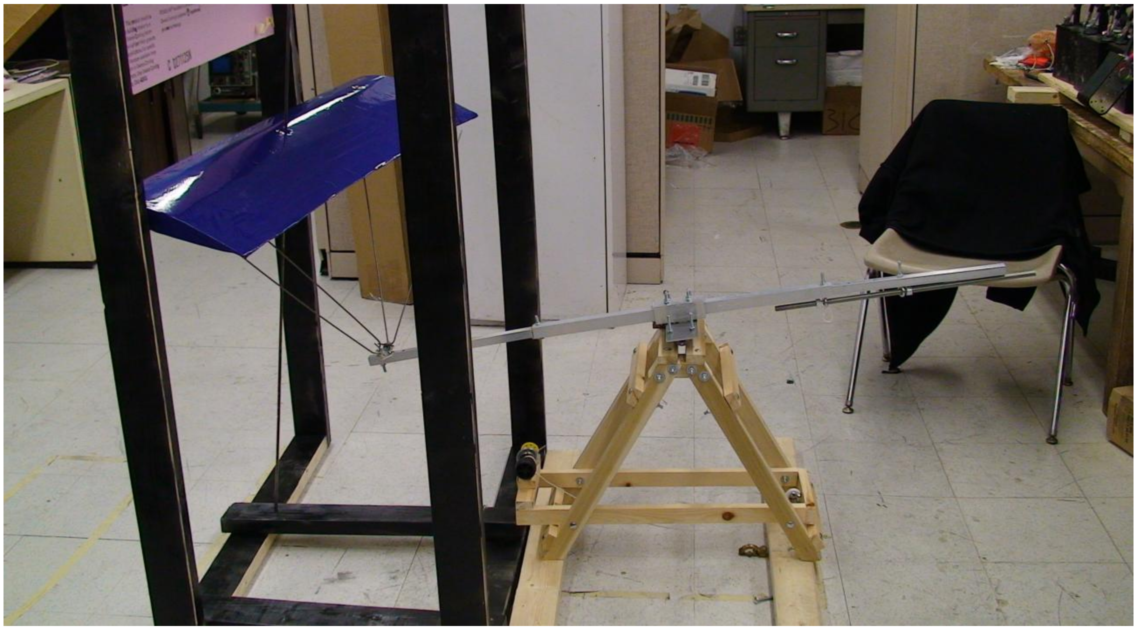

Figure 13: Final Design of the Wind Powered Scale-Model Replica

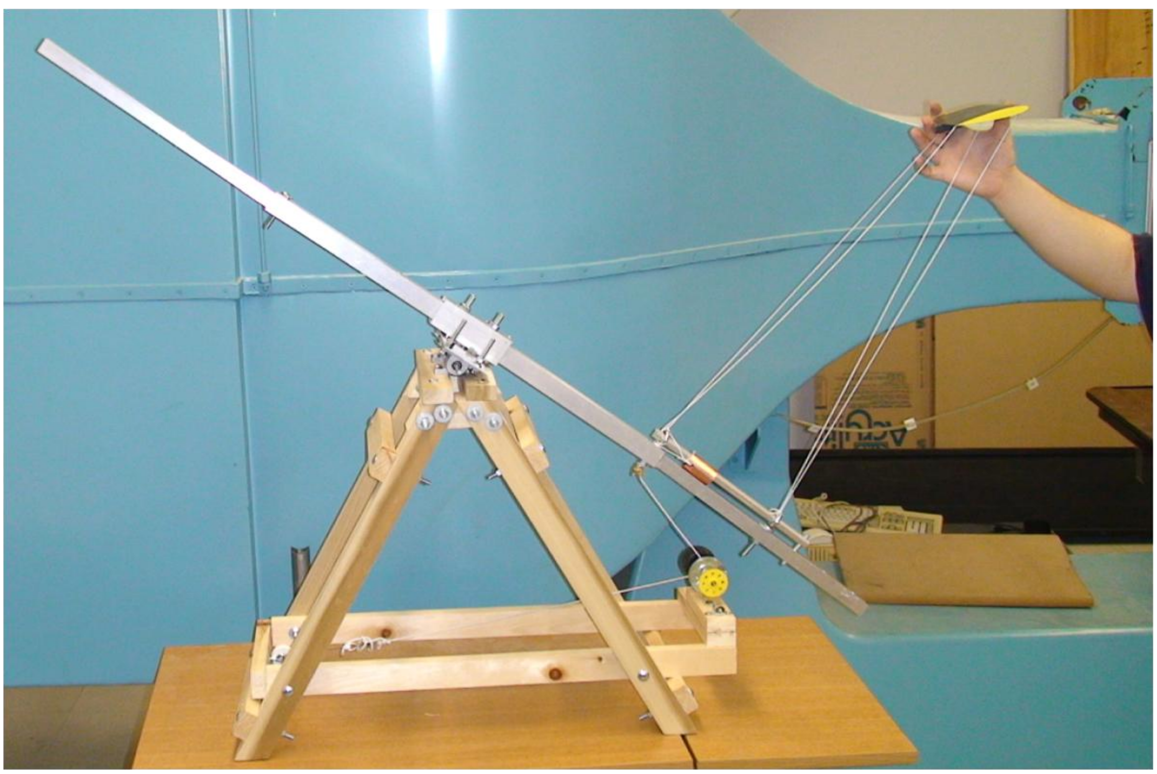

Figure 14: Final Design of the Hand Powered Scale Model Replica

4.2.1 A-Frame Replica

This replica of the A-frame was built at ¼ scale compared to the full scale demonstrator as shown in Figure 13 and Figure 14.

4.2.1.1 Measuring

The first step in creating the scale replica was the measuring of the current full scale demonstrator. The full scale demonstrator was measured using a protractor and a tape measure. The length of all wood pieces were measured and recorded. The angle of the legs relative to the ground was measured using the protractor, and also the angle of any cuts on the model was measured. The number of bolts and their locations on the model was recorded, as well. The number of pillow blocks attaching the arm was recorded, as well as the length of the arm and its individual parts were recorded along with the location of any holes drilled, and the location of bolts through the arm. All of this information was used to make models of all the individual parts in SolidWorks, and created into a SolidWorks assembly. The SolidWorks model of the full scale demonstrator was passed onto the virtual model team for its uses.

With the knowledge of the size of the full scale demonstrator, a geometric scale factor for the model was set. Based on the fact that it would largely be used as a demonstrator in presentations, it would need to be easily moved, and able to fit on a table for easy viewing. A scale of ¼ was decided on for the scale model replica, because it gave a footprint that was a good size for fitting on a table, as well as an arm size that was easy to control while demonstrating. All of the created SolidWorks parts where then scaled down to ¼, and the assembly was remade so that a list of needed parts was easily acquired and could be printed out for material purchasing, and part manufacture. In

Figure 15: SolidWorks Model of A-Frame and Arm, a SolidWorks model of the A-Frame and attached arm with part labels is shown.

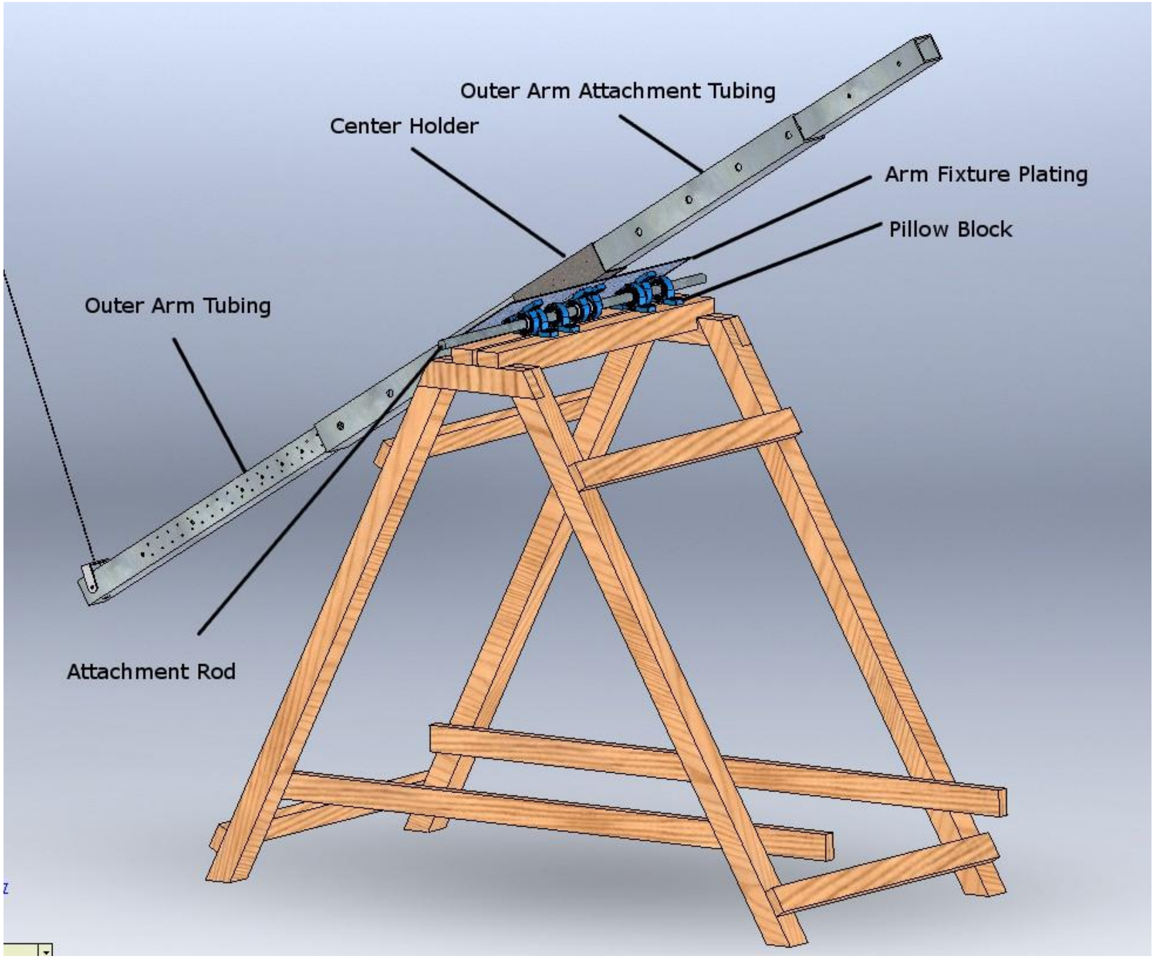

Figure 15: SolidWorks Model of A-Frame and Arm

4.2.1.2 A-frame Material

The A-frame, the wooden A-shaped structure, was constructed from poplar. Poplar was chosen as the construction wood due to its superior looks compared to other available woods, and this was a major point considering the main use for the ¼ scale model was as an educational tool that would be displayed to large numbers of people.

Poplar was also readily available in the smaller sizes needed for construction, and was available at a fair price.

The wood sizes used for the construction of the scale model was 1.5 inch by 1.5 inch wood as replacement for the 4 inch by 4 inch main construction wood of the full scale demonstrator, and 1.5 inch by ¾ inch wood to replace the 2 inch by 4 inch boards used for the cross boards on the demonstrator. These two sizes are a little larger than the 1 inch by 1 inch and 1 inch by ½ inch sizes that the ¼ scale chosen for the model dictated. The exact ¼ scale sizes were not readily available in local stores, and also seemed like they may be too small to give ample stability to the model, as well as too small to avoid splintering when bolts were placed through the wood; the only pillow blocks available in a ¼ inch size, ¼ of the 1 inch used on the full scale model, required the use of ¼ inch bolts for attachment.

The wood was measured out with the use of a measuring tape, a T-square, and a protractor. The lines were marked off with the use of a permanent marker. The wood was then cut with the use of a miter saw for most of the major cuts due to its ease of cutting, especially the angled cuts. A reciprocating saw was also used for four cuts that required cutting only half way into the 1.5 inch by 1.5 inch poplar boards. Some rough edges were sanded as necessary to give a more finished look.

Although the wood sizes were a little out of scale, the actual lengths of all cut pieces were to ¼ scale within reasonable tolerances.

4.2.1.3 Metal Cutting

The metal required metal parts for the creation of the replica rocking arm were cut using a band saw. The arm for the replica was made of 1.5 inch square aluminum tubing as the center piece, 1 inch square aluminum tubing as the piece that the two outer arm parts attach to, and ¾ inch square aluminum tubing was used for the outside two arm parts. 1/8 inch aluminum plating was used for the arm fixture to which the ¼ inch pillow blocks were attached for attachment to the frame. Bolt holes were drilled using a ¼ inch drill bit on a drill press. The cutting positions and drilling positions were marked on the aluminum tubing using a T-Square, a measuring tape, and a permanent marker.

4.2.1.4 Construction

The model was assembled with the use of a drill using a ¼ inch drill but, a socket wrench, ¼ inch carriage bolts, ¼ inch lag bolts, ¼ inch nuts, ¼ inch washer, and wood glue. Carriage bolts were used in places were drilling through the wood was impractical, or would have created a poor appearance, in the fastening of pieces with angle ends, as well as in the attachment of the pillow blocks to the A-frame. Where carriage bolts were being used on the A-frame, holes were first drilled using the drill with ¼ inch bit, and the bolts was put through a washer and nut placed on the end and hand tightened, and then further tightened with the use of a socket wrench. Wood glue was used near the A-frame apex for extra stability.

The arm was assembled using carriage through the holes that were already drilled. The ¼ inch pillow blocks were also attached to the metal plate on the arm. The arm was affixed to the A-frame with the use of a ¼ inch steel rod through the sets of pillow blocks.

4.2.2 Wind Powered Model

In order to help educate the public about the potential for harnessing the wind for energy generation, using kites, a wind powered replica was designed and built. The wind powered model went through much design before a final design was achieved.

4.2.2.1 Parafoil

The wind powered replica used a solid wing to model the kite. A scale kite, scaled at 1:4, would have been too large to use with the fan. The solid wing was cut out of foam. A fan, placed in front of the A-frame and rocking arm, creates the airflow over the wing. The cross-sectional shape of the wing was chosen to be the NACA 4412 (Raymer, 1999). The cross-section is shown in Figure 16.

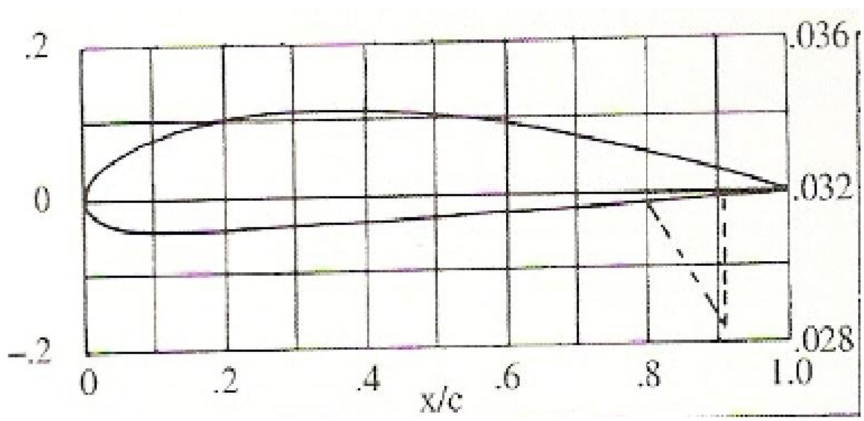

Figure 16: Airfoil Cross-section of Solid Wing

To keep the model in ¼ scale, the solid wing was chosen to be 3x1 feet. The cross-section was blown up using image software to one foot in length. The drawing was printed out and placed over a piece of balsa wood. Using a knife, the balsa wood was cut in the outline of airfoil. This was done two times so that they may be placed on either sides of the foam. A piece of 3 feet by 1 foot insulation foam was purchased and we attached the balsa wood. In order to get a smooth cut, a heated wire was used. The wire was pulled through the foam in the shape of the wing. Ultracote, a coating commonly used for model airplanes, was then wrapped around the parafoil. The coating was placed on the foam, and hot air was blown on the coating to get it to stick to the foam. A picture of the final solid wing is shown in Figure 17.

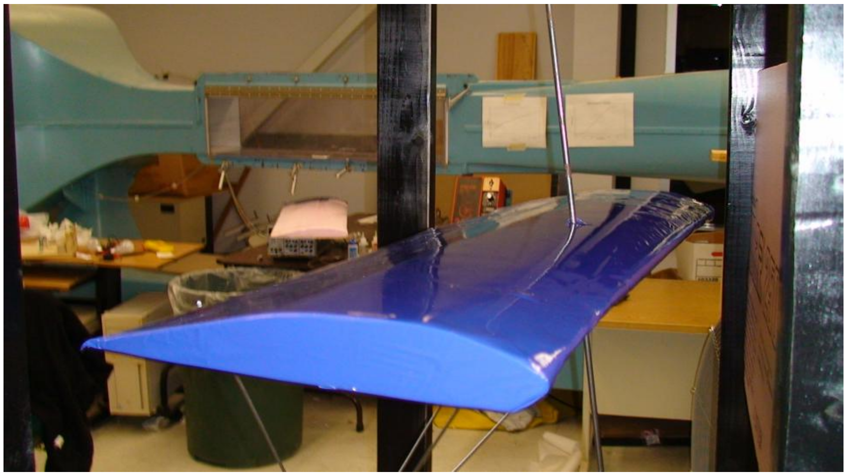

Figure 17: Finished Solid Wing

4.2.2.2 Attaching the Solid Wing to the Replica

At first, the parafoil was attached rigidly to the scale A-frame using a single metal rod. This proved to be very problematic because the angle of attack would change radically as the arm was raised and lowered (see Figure 18). The dramatic change in angle of attack prevented the arm from falling back down.

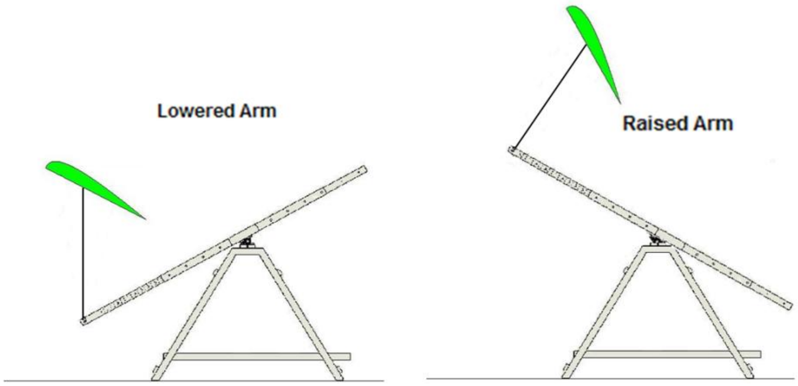

Figure 18: Difference in Angle of Attack in Upward and Downward Position

We needed a way to control the angle of attack in a way that allowed the angle of attack to stay constant throughout the rocking motion of the arm.

Attaching a hinge to the point where the metal bar attached to the arm would allow the parafoil to maintain a more constant angle of attack. Also, to prevent the foil from twisting and turning, four thin metal rods were attached to the corners of the parafoil. Figure 19 shows the parafoil attached with four support rods.

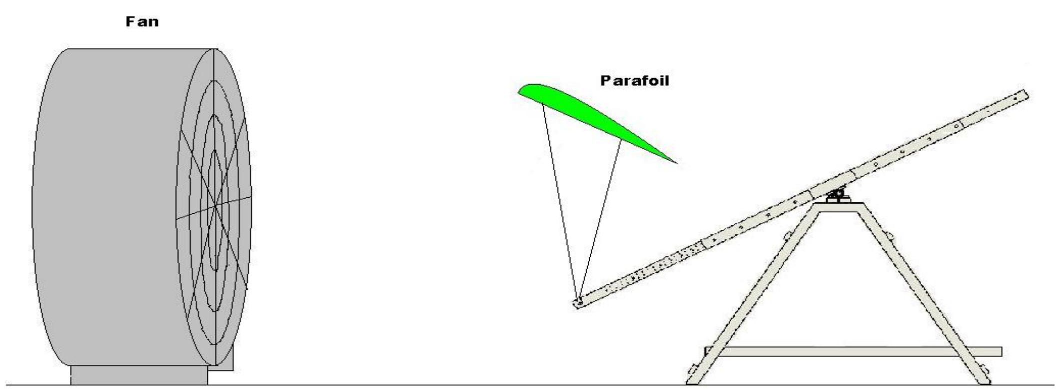

Figure 19: Second Iteration of Wind Powered Replica

This configuration had a few problems. The parafoil had a tendency to fall towards the arm due to the weight of the parafoil and supporting rods and the fan was not strong enough to raise the arm. In order to prevent the parafoil from falling over, and to maintain a constant angle of attack, a support structure was constructed. The structure has a tall metal rod that the parafoil slides up and down on. The structure was built out of 2x4 inch pieces of wood and screws. The rod prevents the parafoil from falling over and helps maintain a constant angle of attack. In addition, the structure has a piece of foam that stops airflow when the arm is in its upper position, allowing the parafoil to depower and fall back into the airflow. A ½ inch threaded rod was added to one of the arms for use as a counter-weight in the scale models use as a wind-powered system to offset the weight of the parafoil and parafoil supports. Eye hooks were added to one of the arms, eyes to the bottom on the arm, and affixed with ¼ inch washers and nuts. The ½ inch threaded rod was then slid through, and two ½ inch nuts were used to hold it firmly in place, while still allowing for later adjustments depending on parafoil and support weight. Figure 20 and Figure 21 show the schematic and the final wind powered replica.

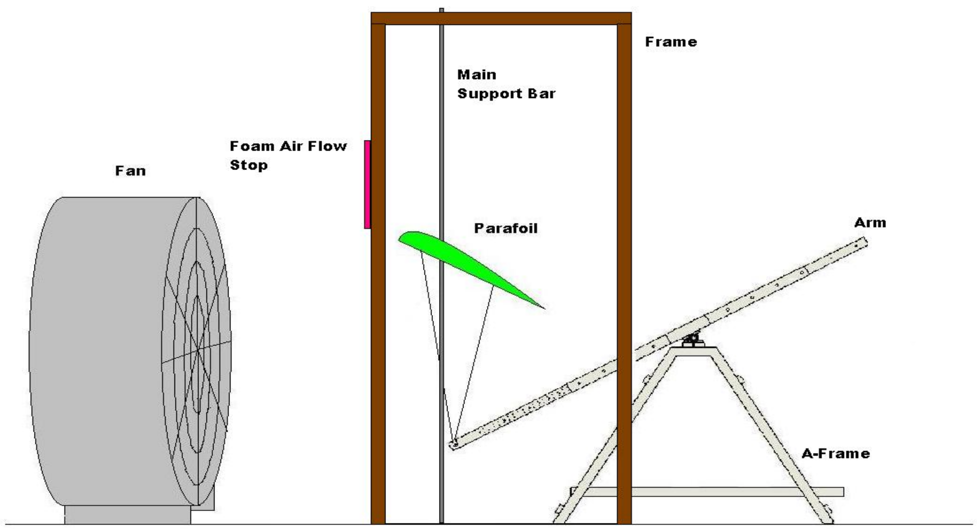

Figure 20: Schematic of Design of Wind Powered Replica

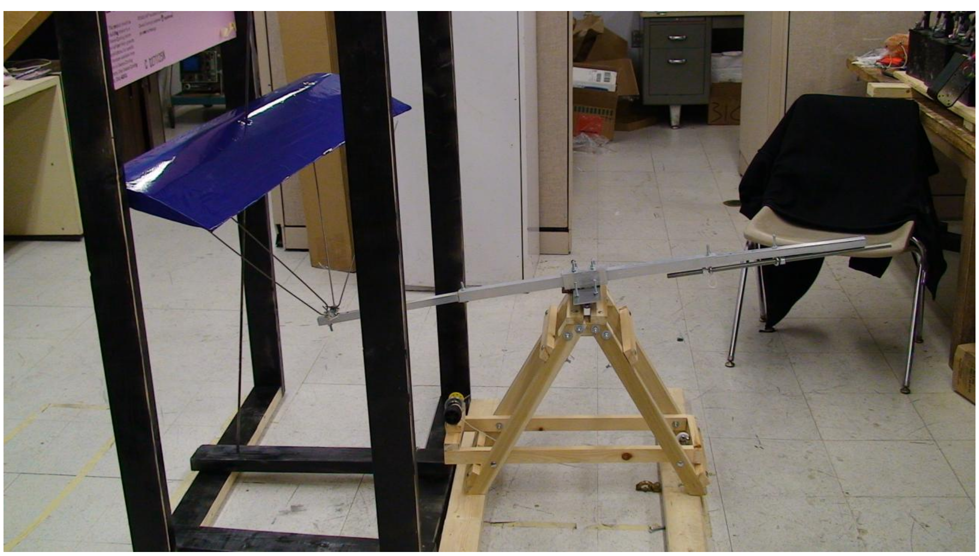

Figure 21: Final Design of Wind Powered Replica

4.2.3 Hand powered model

A hand powered replica was developed to provide a simple educational tool that clearly shows how the kite power demonstrator operates. Several additional components were developed and attached for the hand powered replica. These additions were designed to be easily removable from the model, as to allow for easy transportation of the model and a simple conversion process back to the wind powered set up.

4.2.3.1 Angle of Attack Mechanism

To mimic the angle of attack device on the full scale demonstrator a simple sliding system was created. After measuring the full scale's sliding weight system to be approximately 40 inches long with a 12 inch slider, we cut a ¼ inch diameter aluminum bar to 10 inches long and purchased a 1 inch long, ¾ inch diameter copper pipe.

This system was then mounted onto the base model's arm. This was done by first attaching two metal eyes to the pre-drilled holes in the model and then sliding the ¼ inch diameter bar though both holes. The copper slider was put on the bar between the two eyes. To fully secure the bar to the arm, a hole was first drilled into the side of the bar. After this a metal peg was inserted in the hole and then into a pre-drilled hole on the arm. See Figure 22 for a picture of this system.

Figure 22: Angle of Attack Mechanism

To mimic the kite itself we used a small, 4 inch by 6 inch plastic wing for the hand powered model (See Figure 23). While not to scale, this kite model effectively demonstrates the kite motion. A scale model kite would be approximately 3 feet by 1 foot. Such a large kite would be hard to hold, and thus would make demonstrations difficult. The kite is attached to the base arm using 4 pieces of string modeling the 4 control tethers of the kite power demonstrator. Two pieces of string attach the leading edge of the kite to the front metal eye, and the other two pieces of string attach the trailing edge of the kite to the sliding black steel coupler.

Figure 23: Hand Powered Model Kite

4.2.3.2 Power Train

A simple power train was constructed to model the power conversion mechanism (sprag clutch, retractor spring, gears, flywheel, generator) of the full-scale demonstrator (see Figure 24). We assembled basic pieces from an Erector set into a simple system with a low speed shaft connected to a high speed shaft though a 3:1 gear ratio. We then mounted this system on the replica A-Frame. Several flywheels were also added to this system.

A common keycard retractor was then connected to the opposite side of the base model. A string was then glued to the arm above the shaft system. This string was then wrapped around the shaft several times, and then finally connected it to the retracting end of the keycard retractor.

A permanent magnet DC generator was then connected to the end of the high speed shaft. This was done to allow our system to generate electricity. A voltmeter was then attached to the motor to help demonstrate the power generation ability of the system. This system can be seen in Figure 24.

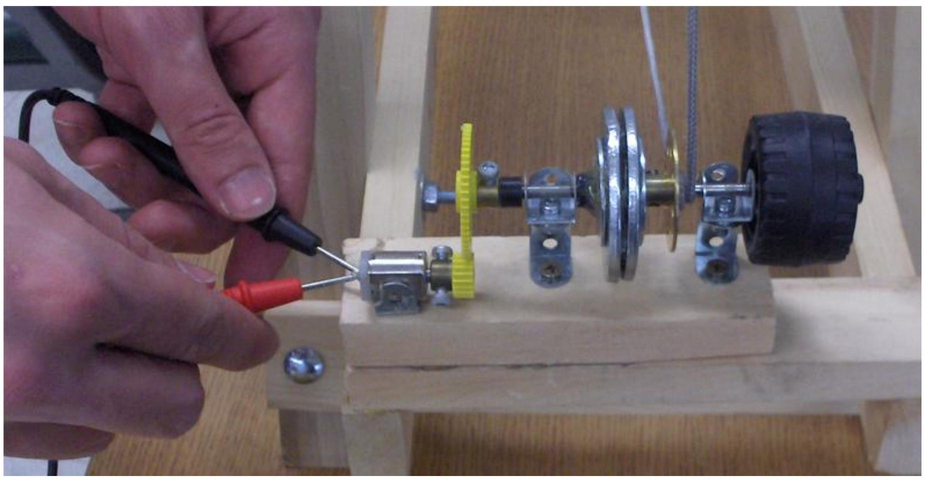

Figure 24: Scale Model Power Train

4.3 Virtual Animation

We developed a virtual animation of the full-scale kite power demonstrator. It can be used when it is unreasonable to display the actual full-scale kite power demonstrator. It is also perfectly suited for display on a website or over the internet. It also helped our project team understand the kite motion and dynamics of the real kite power demonstrator. It is used in conjunction with a MATLAB simulation developed by Olinger and Goela (2008) which provides the input data to the virtual animation. The individual parts were modeled in the CAD program, SolidWorks, and the assembly was animated in the add-on, SolidWorks Animator. Figure 25 shows two screenshots of the virtual animation.

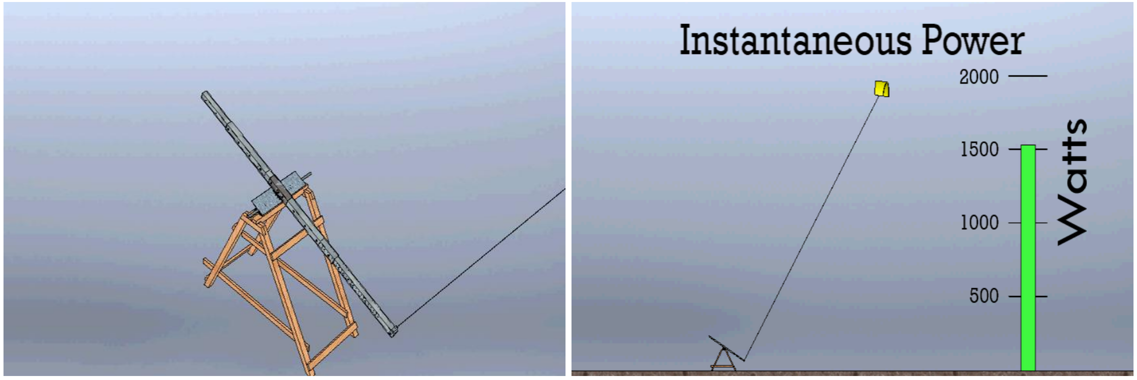

Figure 25: Screenshots of Virtual Animation

As input, the virtual animation uses the dynamic time history of the kite motion, tether angle, rocking arm angle, instantaneous power, tether tension, and lift to drag forces on the kite, obtained from the simulations of Olinger & Goela (Goela, 2008). These simulations are run in MATLAB, and they model the physics and dynamics of the complete kite power system. Some of the equations included in the MATLAB simulation can be found in Figure 26: Physics and Dynamics Equations. More information regarding the calculations for the simulation can be found in "Performance Characteristics of a One-Kilowatt Scale Kite Power System," written by Olinger and Goela. The entire MATLAB code can also be viewed in Appendix A. The virtual animation shows the movement of the arm, tether, and kite when the kite power demonstrator is in motion. A simplified representation of the demonstrator's power conversion mechanism was also animated. The virtual animation can be found on the project wiki site at this web address: http://www2.me.wpi.edu/wpi-kites/index.php/Virtual_Animation.

$$\frac{dV_{2A}}{dt} = \frac{g}{W_K}\left(F_{DK}\sin(\theta+\phi) - F_{LK}\cos(\theta+\phi) + W_K\cos\theta \right) \qquad \frac{d\theta}{dt} = \frac{-V_{2A}}{L_t}$$

$$\frac{dV_A}{dt} = A\begin{bmatrix} F_t R_A \cos(\gamma - \theta + \pi/2) - W_{LOAD}\dfrac{R_{G2}}{R_{FG}}\dfrac{R_F}{R_G}R_C \\ -K\Delta x R_C - M_{AC} \quad -W_{BA}\dfrac{R_A}{2}\cos(\gamma) \\ +W_{DB}\dfrac{R_D}{2}\cos(\gamma) + W_{CTR}R_{CTR}\cos(\gamma) \end{bmatrix} \qquad \frac{d\gamma}{dt} = \frac{-V_A}{R_A}.$$

$$I_F\frac{d\omega_F}{dt} = -W_{LOAD}R_F.$$

Figure 26: Physics and Dynamics Equations

4.3.1 Modeling Difficulties

A complete list of the sub-components in the virtual animation can be found in Appendix B. Most of the demonstration sub-components were straightforward to model and assemble. The one sub-component that was most difficult to model was the kite. Figure 27 shows the dimensions of the 10 meter squared Peter Lynn Guerilla kite used in the full-scale demonstrator as it lies flat. Approximate reference planes created in SolidWorks to model an accurate kite are also shown. Some liberties had to be taken in order to make the kite curved, but overall it is consistent with the actual kite dimensions.

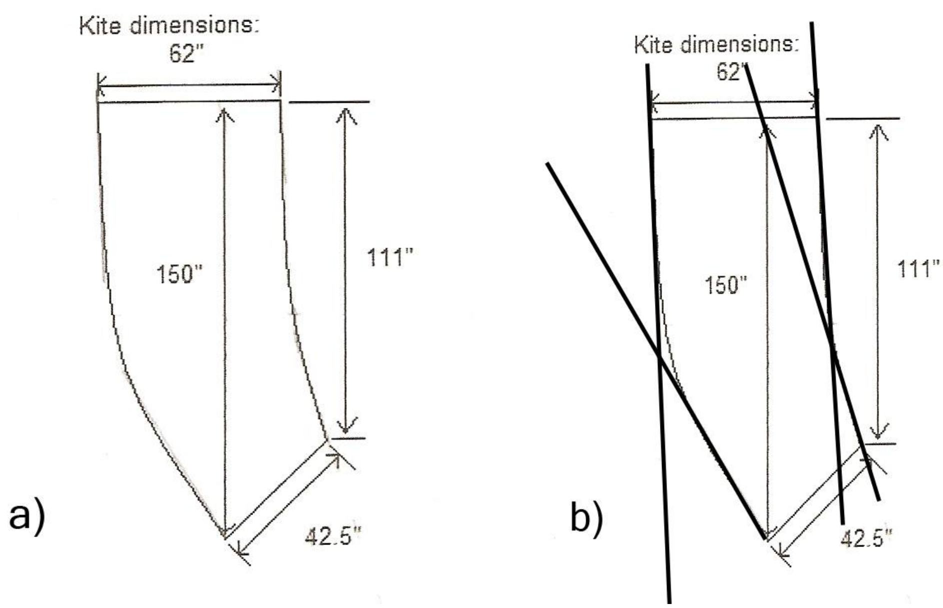

Figure 27: Virtual Model Kite Dimensions: a) Dimensions of the Peter Lynn Guerilla kite used in the full-scale demonstrator b) The kite's dimensions overlaid with reference planes used to make the part in SolidWorks.

4.3.2 Modeling Revisions

The SolidWorks assembly of the mechanism has been continuously updated throughout the project. Occasionally, drastic changes were made to the assembly, such as when the original "filler" kite model was replaced with the more accurate version. However, the "filler" kite model was involved in several important geometric relations that were necessary for the virtual animation to function properly. In such cases, the old models were simply hidden by the program, so the geometric relations remained, but the model did not. Even though these pieces do not appear in the animation, it is important that they remain in the assembly to maintain these geometric relations.

4.3.3 Exporting Data from MATLAB

The MATLAB simulation contains complex mathematical calculations that simulate the actual physics of the kite and mechanism. More specific information on this simulation can be found in Dr. Olinger and Dr. Goela ASME Energy Sustainability 2008 conference paper. The simulation calculates many different variables and stores them with their corresponding timestamp, so any piece of data can be obtained for any time. For animation purposes, we need: arm angle, tether angle (with respect to the horizontal), kite angle-of-attack, instantaneous power, tether tension, and lift to drag forces on the kite. By specifying the variables needed, MATLAB exports all the data into a Microsoft Excel spreadsheet where the data can be observed and input into SolidWorks. Appendix A contains the entire MATLAB code.

4.3.4 Animation Process in SolidWorks Animator

The sub-components are all put together in an assembly by specifying appropriate geometric constraints. Most of the relations are made to hold the mechanism together, but some of them are responsible for the animation. The constraints important to the virtual animation are: Gamma, Theta, New Alpha, Power, Tension, and LDPlot. All the remaining constraints are for holding the mechanism in place as these six relations change.

The six important relations all correspond to pieces of data exported from MATLAB. Gamma contains the angle of the rocking arm in relation to the ground. Theta contains the angle of the kite tether with respect to the horizontal. New Alpha contains the angle of attack of the kite. Power, Tension, and LDPlot each controls how far their corresponding bar rises.

Due to the way some of the geometric constraints are defined in SolidWorks, the exact numerical data cannot be taken straight from the MATLAB simulation. The data must usually be transformed in some way to account for the geometry. Appendix C has the Excel spread sheet information that contains the values for Gamma, Theta, New Alpha, Power, Tension, LDPlot, and the data for the power conversion mechanism. The value for Gamma exported by the MATLAB simulation must be subtracted from 90 in order to get the proper numerical value to input to SolidWorks. The value for Theta exported by the MATLAB simulation does not need to be transformed to get the proper numerical value to input to SolidWorks. The value for New Alpha exported by the MATLAB simulation must be added to 90 in order to get the proper numerical value to input to SolidWorks. The value for Power exported by the MATLAB simulation must be divided by 2 in order to get the proper numerical value to input to SolidWorks. The value for Tension exported by the MATLAB simulation must be divided by 3 in order to get the proper numerical value to input to SolidWorks. The value for LDPlot exported by the MATLAB simulation must be multiplied by 100 in order to get the proper numerical value to input to SolidWorks. The value for OmegaG of the power conversion mechanism determines whether it is in the engaged or disengaged state (engaged if OmegaG is positive, disengaged if OmegaG is negative). When engaged, the value of the angle input in SolidWorks decreases by increments of 45 every tenth of a second, and when disengaged, the value of the angle increases by increments of 45 every tenth of a second. The remaining gears in the power conversion mechanism do not change direction, and are decreased by increments of 45 every tenth of a second.

The animations were done by making key frames at a set interval, either every quarter of a second or every tenth of a second. The data exported from MATLAB was entered into the appropriate geometric constraint at the appropriate timestamp in SolidWorks. Figure 28 shows all the key frames in the twenty second animation.

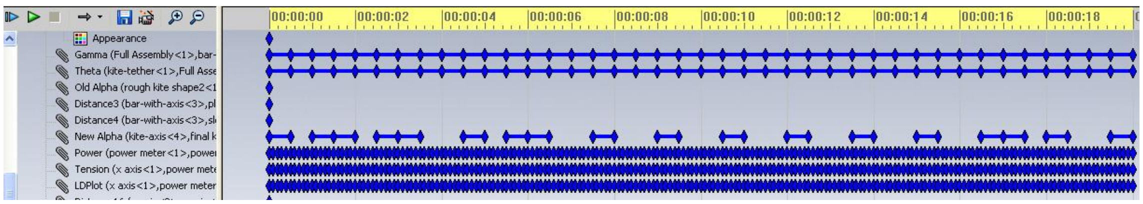

Figure 28: SolidWorks Key Frames

Each blue diamond is a key frame that corresponds to calculated MATLAB data.

4.3.5 Cameras

One of the early concerns of the virtual animation was how to choose the best camera for viewing the animation. The camera could not move or the viewer would become disoriented.   If two cameras were used in conjunction, their scale had to be exactly the same to maintain accuracy of the model. The end result was to simply have the animation rendered for several different camera locations and make them viewable in a single video. This way the viewer can experience the animation from many locations, and can better understand how the kite power mechanism works.

4.3.6 Directly Importing Data from MATLAB to SolidWorks

One of the most important features of the virtual animation is that it runs off of simulated MATLAB data. In the early state of the project, there was an idea to try to import the data straight from MATLAB into SolidWorks Animator and have all the data input automatically. This turned out to be beyond our capabilities, and instead we used Microsoft Excel as an intermediary between the two programs. Also, the data had to be put in manually for each key frame.

4.4 The Electrical System

Once the kite power demonstrator has harnessed the power of the wind and converted it into mechanical energy it will need to be turned into usable electrical energy. The goal of the electrical designed system is to provide the means for transferring the mechanical energy into electrical energy that can be stored and used as needed. It has also been designed so that it can help demonstrate to the general public though its display at Overlook Farm how providing electricity to remote areas provides for the needs of people in developing nations. To demonstrate the electrical uses of people in the developing world it was imperative to provide a physical example of what could be powered by the system. People from different developing nations have different electrical needs. Therefore we developed a flexible electrical system that can power a variety of devices including light bulbs, water pumps, TVs, laptops, radios, and communication devices including cell phones. A display of these loads accompanied by the electrical system and relevant information will effectively provide a comprehensive educational tool.

In order to determine the optimal electrical system for the developing world we analyzed traditional small scale alternative energy electrical systems and then analyzed design characteristics necessary for an effective system. We also utilized pros and cons for each component of the system to determine the best choice. Finally we considered the cost of entire system.

4.4.1 Overview of System

The system we developed (Figure 29) has quite a few components that provide the functionality that is required to provide a demonstration of different uses of electricity. The key functions are to store electricity, allow electrical charging, provide AC electricity, and prevent damage to the system and users. Electricity is stored in the battery bank. The DC generator creates electricity from the kite demonstrator's rotating shaft to charge the batteries. The inverter converts the DC electricity into AC electricity and the circuit breakers distribute the electricity to needed electrical loads. Damage to the system and dangerous conditions are minimized by using undercharge protection, overcharge protection, fuses, circuit breakers, and grounding. Undercharging the batteries can lead to shorter life for the batteries and is prevented by using an inverter that will not continue to convert DC to AC electricity if the batteries display a too low charge. Overcharging the batteries can lead to explosions and the release of toxins from the batteries thus destroying the batteries and creating a dangerous environment for users. Fuses and circuit breakers contained in metal boxes, provide protection from short circuits and excessive loads and therefore protect the electronics from damage and the user from electric shocks. Grounding wires run between all the electronics and the ground allow a safe route for electricity to flow if a short circuit thus preventing damage to the electronics and the user from an electrical shock.

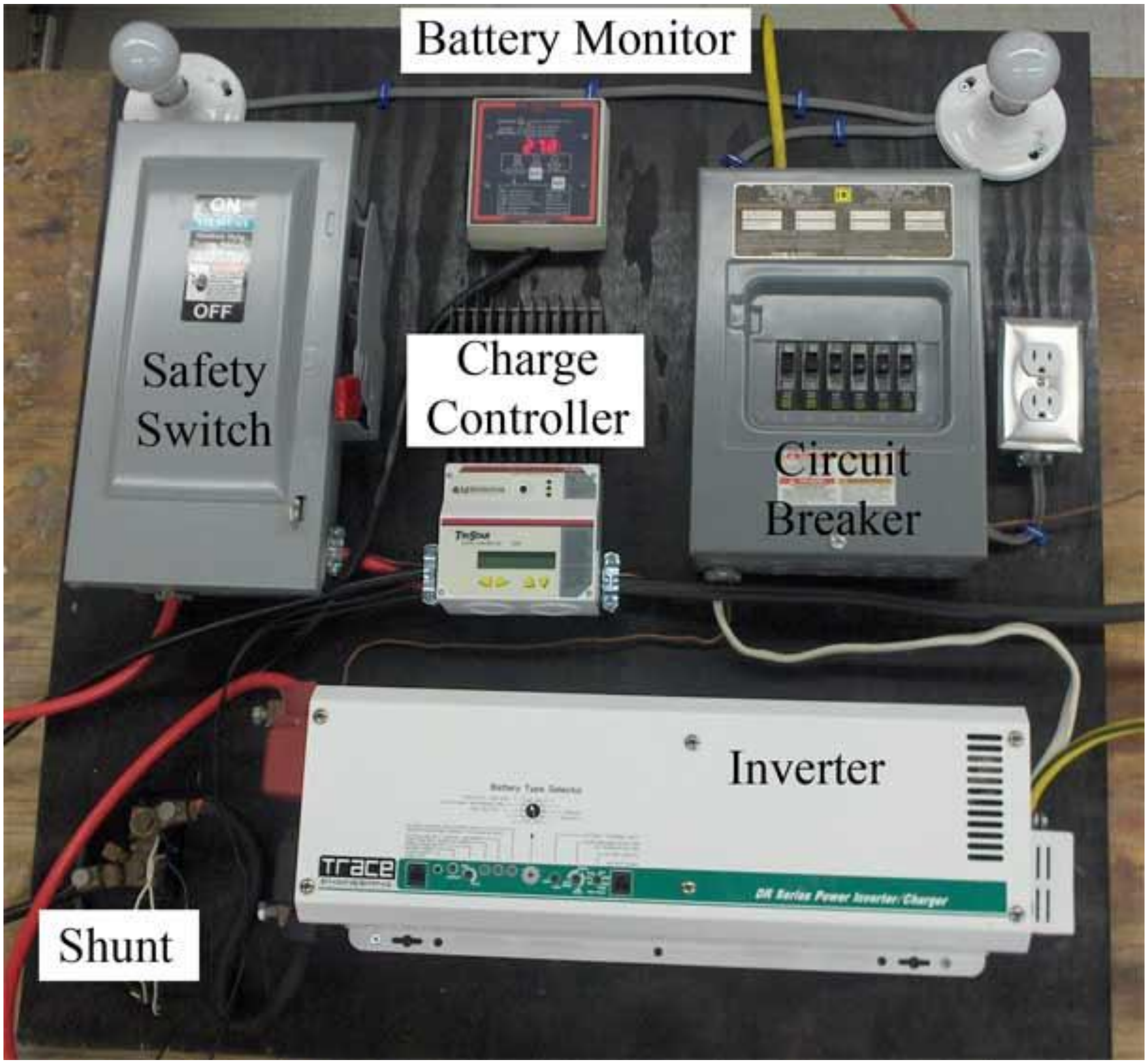

Figure 29: Overview of Electrical System

The electrical layout of the system can be seen in Figure 29, which presents the flow of electricity in the system. DC electricity flows from the generator to the batteries with a plug that can cut off the electricity flow if desired. In the first of step of the primary system, where usable AC electricity is created, DC electricity flows between the batteries and inverter that converts it into AC electricity. There is a shunt, a precise and extremely low level resistor, on the negative cable between the batteries and inverter that allows the usage of a meter to measure useful information about the system such as the voltage of the batteries, amps used by loads, and percentage of battery charge remaining. The one-phase AC electricity that the inverter creates flows to a AC circuit breaker box where individual breakers can turn on and off electrical distribution to various electrical loads thus completing the primary system. In the secondary system electricity flows from the batteries through a fuse box to a charge controller. The charge controller acts as the facilitator for the diversion load, when it senses an excess electrical charge in the batteries it turns the load on and when the charge has lowered to an acceptable level, it turns it back off. When the diversion load receives electricity it transfers the DC electricity into heat energy using heating elements that raises the temperature in the water tank. For a schematic of the electrical layout of the system see Figure 30.

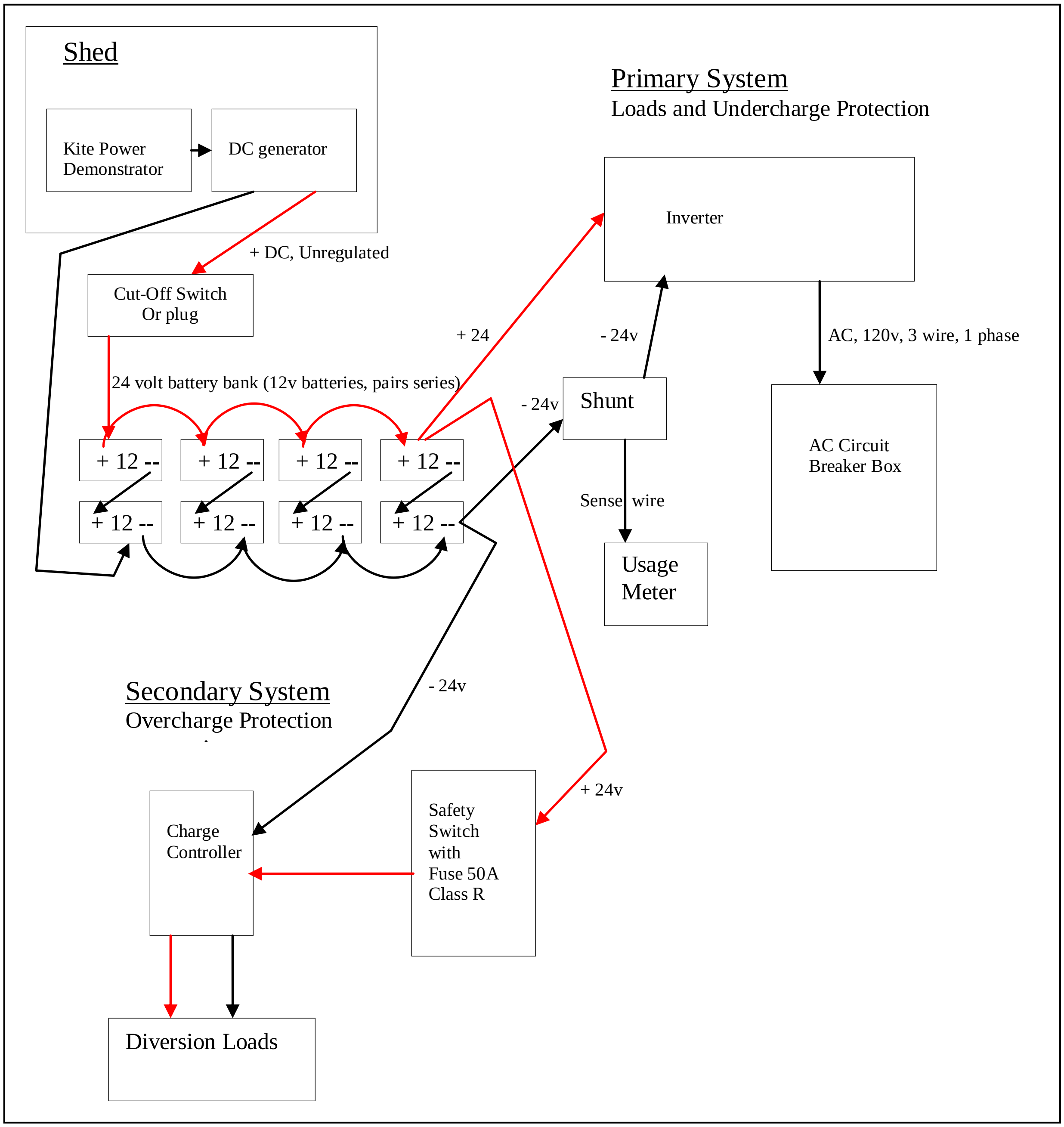

Figure 30: Electrical Layout

4.4.2 Analysis of small scale alternative energy electrical systems

We analyzed traditional small scale alternative energy electrical systems such as the system that IQP students built at Overlook Farm in 2000. We looked at all facets of the design of the systems and how the students determined various components and capacities. Paying particular interest to characteristics that are necessary for an effective system we could determine how our system should differ from typical wind turbine systems in order to develop the most effective system. In fact we were able to utilize some of the components from the wind turbine and solar electrical system that the IQP built at Overlook Farm in 2000. The wind turbine and solar electrical system is no longer operational at Overlook farm since its source of electrical generation had been destroyed by a fire. In the end we were able to use the inverter, battery monitor, shunt, circuit breaker box, battery terminal connector cables, and various other wires. One difference our system has compared to the previous Overlook Farm electrical system is their system was designed to power a powerful water pump for long durations and because of this design feature they require a lot larger electrical storage capacity. Our electrical system is very similar to other alternative energy systems however our power usage goal is broader than the average system so more capabilities must be built in.

4.4.3 Determination of Electrical Storage Capacity

In order to determine the electrical capacity for our system we first analyzed what uses in the developing world required and would use. We then created a system with enough capacity to meet these energy needs for extended periods of time to allow the ability to demonstrate many scenarios. We decided that the electrical loads that we would want to use would need approximately 600 watts for about 5 hours or 300 watts for about 10 hours depending on the magnitude of the load. From this we could determined that the required watt-hours was a little over three thousand. From the watt-hours we calculated the amp-hours within the desired 24 volt system to be 130 amp-hours. Amp-hours are the standard way to measure how much electricity is stored. To power all the lights, TVs, laptops, and other loads we will need a battery bank with a storage capacity of 130 amp-hours.

4.4.4 Component Analysis

The kite system will be located at Overlook farm and because of its staffing limitations the system must remain functional for long durations with minimal user assistance and especially without skilled user assistance. To achieve these ends we had to select electronic components that would individually and as a whole fulfill these requirements.

There are many valid options for the different components of the system. To determine which product to use we looked at the advantages and disadvantages for each option and then weighted them appropriately to determine which option we would choose.

4.4.4.1 Battery bank

The electrical energy must be stored in a battery bank. There are many factors that play into determining the type and quantity of batteries. We determined the amount of volts that would be the optimal system voltage. In addition we determined the desired amount of electricity storage capacity needed from the power consumption of components over a period of time. We laid out a comprehensive table (see Table 4) showing the differences in various choices of batteries. Using this table we were able to identify the ideal battery choice, which were the Deka/MK AGM type batteries (Figure 31) with 12 volt potential and 32.5 Ah capacity per battery with a total of 8 batteries comprising the bank.

Table 4: Battery Comparison

volts	Ah per bat.	#bat. in string	#bat, total	$/unit	total cost	name
12	50	2	5.2	120.2	625.196	Universal Ub12500 12V, 50Ah (20Hr) Sealed Agm Bat
12	34	2	7.65	131.9	1008.8	Concorde Pvx-340T 12V, 34Ah (24Hr) Sealed Agm Bat
12	42	2	6.19	138.7	859.1762	Concorde Pvx-420T 12V, 42Ah (24Hr) Sealed Agm Bat
12	75	2	3.46	184.9	641.2293	Universal Ub12750 12V, 75Ah (20Hr) Sealed Agm Bat
12	100	2	2.6	203.2	528.32	Universal Ub121000 12V, 100Ah (20Hr) Sealed Agm
12	73	2	3.56	191.1	680.737	8G24 12V, 73 AH (20HR) Sealed Gel Cell Dual Terminal
12	97.6	2	2.66	218.4	581.8033	E31SLDG 12V, 97.6Ah (20Hr) Sealed Gel Cell w/ Flag Terminal
12	100	2	2.6	338	878.8	Trojan AGM 27 - 12v - 100Ah
6	220	4	2.36	155.1	366.8127	Trojan flooded lead-acid 6v T105
12	32.5	2	8	68.34	546.72	Deka/MK Battery 8AU1H (T873) AGM, 12 volt 32.5 Ah - Chosen Battery Type

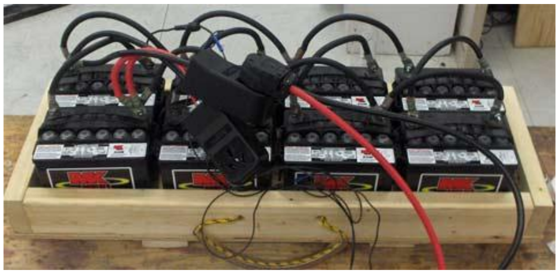

Figure 31: Deka/MK Batteries

4.4.4.2 Generator

The generator, or alternator, transfers the mechanical energy of the kite arm mechanism into electrical energy. It transforms the rotational energy of the rotating shaft and turns it into electricity. We looked at several choices for the generator. Some rectify the electricity directly into a form that is usable and others require additional components. Cost and capacity were the major functions in determining the choice. After comparing several alternators we were able to determine which would be the best choice for this project. Windblue's model 512 permanent magnet alternator was chosen as it can produce the desired electrical output of one kilowatt of electricity within adequate rpm values (~2000) and has a very reasonable price. The Windblue generator's voltage and amperage output compared to its rpm values are shown in Figure 32.

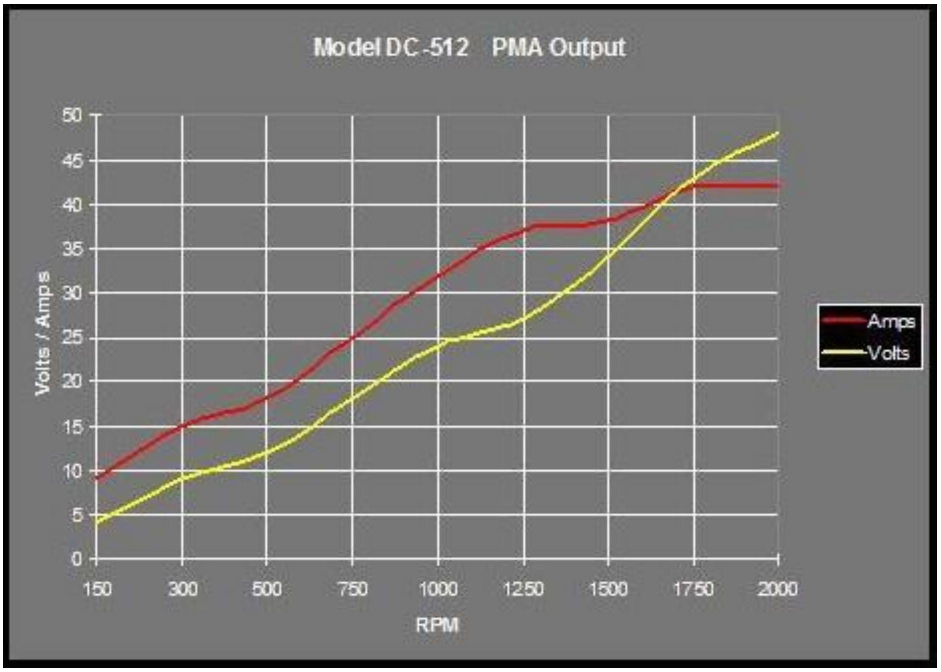

Figure 32: Windblue's Generator Model 512 - Voltage & Amperage / RPM

4.4.4.3 Over-load Controller

When the charge on the batteries becomes excessively high a load must be applied in order to lower the charge and avoid battery damage. We looked at several different charge controllers and found one that performed the necessary functions at as fit a reasonable cost. Morningstar's Tristar Ts-60 fit our needs best since it had the necessary capacity of transferring 60 amps, the lowest price, and most flexibility in functionality.

4.4.4.4 Inverter

The inverter converts DC electricity into usable AC electricity on demand. These machines are very complex and can have a wide range of costs. We analyzed several different choices as well as determining if the one provided from Overlook Farm could fulfill our needs and eliminate the necessity to purchase one. Since Overlook Farm is partially funding the project it would decrease their overall monetary investment if they donate components. In the end we used the Xantrex Trace 3624, provided by Overlook Farm. It has a capacity to provide 3600 watts of electricity continuously. While this is higher than our needs it has many functions that less expensive inverters would not necessarily have such as undercharge protection.

4.4.4.5 Electrical Loads

The electrical loads that we determined to use for the electrical system to demonstrate the power output include two light bulbs, an old fashioned TV, and a laptop. However there are countless other possible electrical loads that could be used with the system as any electronic gadget with an AC plug can be used with the system. There are currently three AC plugs available to plug AC loads into but more could be added easily if needed.

4.4.5 System Cost

To maximize its potential usage in the future, the kite system must produce reliable and competitively priced electricity for its applications. It will require minimal cost in both the physical construction and long term sustainability. We were able to obtain some components without cost because they were donated from Overlook Farm. Energy usage for the future of the system was looked to determine the scale of generation and storage and capacity needed which determined what consumptive devices could be used and how much they will be used. Because of predetermined power requirements we had to scale the system, and thus cost, to the capacity necessary to fulfill those requirements. The overall electronic system cost with individual costs displayed is shown in Table 5.

Table 5: Overall Electronic System Cost

Item	Cost
Windblue DC-512 Motor or Wind driven Permanent Magnet Alternator	$199.00
2 Gauge AWG Battery Wire cable 20 ft, black and red each	$65.95
Heating elements - Two pack of #SJH126002 - for 12 volt systems- 7" model	$77.85
*Inverter - Trace 3624	$1,195.00
*Batery Monitor - Trimetric Monitor	$194.00
*Wiring - #2 gauge battery connectors and between other components	$199.75
*500A-50mV Shunt	$21.50
*AC Circuit Breaker Box	$83.00
50 Amp, 250 VAC/ 125 VDC Class R Fuse	$50.90
Deka/MK Battery 8AU1H (T873) AGM, 12 volt 32.5 Ah	$691.72
Morningstar Tristar Ts-60, 60A Chg Ctlr	$184.00
Morningstar Tristar Dm Digital Display	$90.00
Battery Terminal Connector 3/8" UL Listed Antirotational Lug, 2 Wire Gauge x10-7106K94	$48.30
Safety Switch: 2 DPST, 3-Wire 60 10 15.7" x 6.6" x 5.1" 7524K22	$81.45
Miscellaneous components such as plywood, wire, connectors, loads, etc.	$350.00
*Acquired from Overlook	
Total of System	$3,532.42
Overlook has contributed	$1,693.25
Funds used	$1,839.17

4.5 Website

The website was created using the MediaWiki framework and was hosted on servers on the WPI campus. A Wiki format was chosen for the website for its relative ease of creation. An understanding of the code that could be used in the creation of the Kite Power wiki was gained largely through the reading of the source code for a number of pages on Wikipedia. Some of the Kite Power Wiki's code was taken from Wikipedia pages, as this is allowed because all code hosted on the MediaWiki framework can legally be used by other parties under the GNU General Public License.

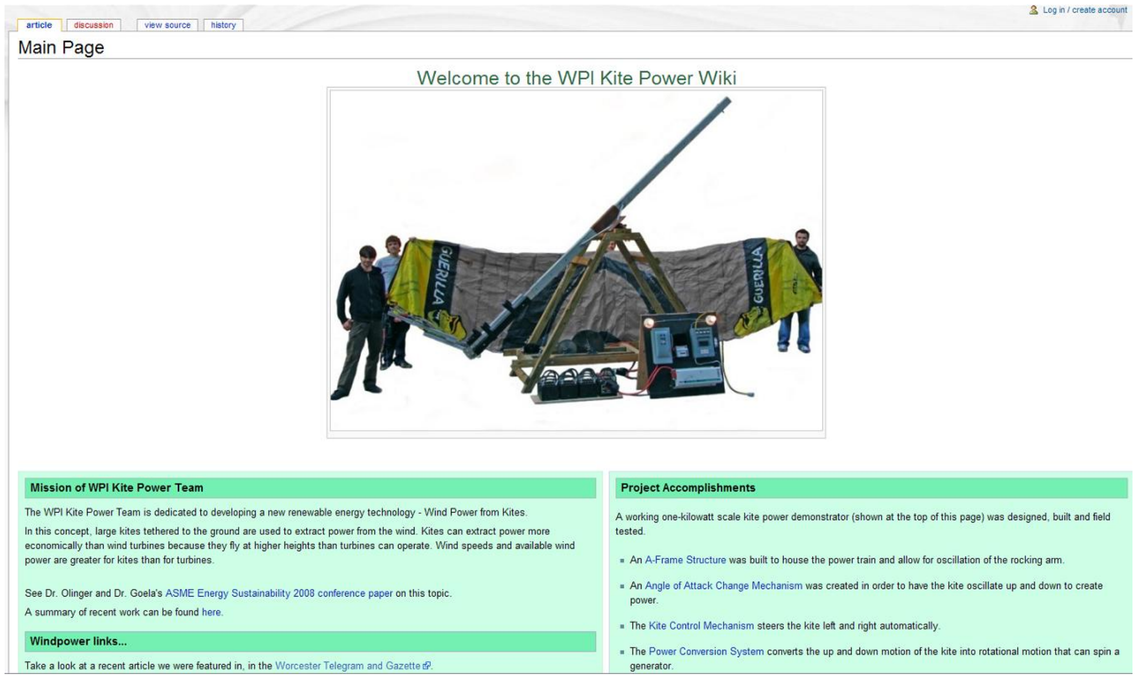

Figure 33: Main Screen of Kite Power Wiki

Text and images for the site were acquired from members of both the MQP and IQP group working on the Kite Power project. Complementary colors were chosen for a pleasing look on the main page, and images were resized to achieve a balanced look on the pages using the open-source image editor GIMP for its easy resizing, as well as the sustaining of good image quality through its scaling(see Figure 33 for a screenshot of the main page of the wikisite). A green color scheme was selected to reflect that kite power is a "green" or renewable energy. Videos were posted on YouTube, and linked to from the Kite Power Wiki to allow users to get a more in depth idea of how many of the systems worked in to better educate them. The link to the WPI Kite Power Team wikisite is http://www2.me.wpi.edu/wpi-kites.

5. Results and Conclusions

5.1 Scale Model Replica

The final scale model replica shown in Figure 14 is an effective small-scale demonstrative tool. Both the wind and hand powered replicas illustrate various elements of the full scale demonstrator.

5.1.1 Wind Powered Set up

The wind powered set up of the scale model replica (Figure 13) demonstrates the ability of the system to operate with wind as its only input. This model effectively harnesses wind power and translates it to up and down oscillation of the kite arm. The model does, however, have a limited arc of movement, and it requires an external support structure.

5.1.2 Hand Powered Set up

The hand powered set up of the scale model replica (Figure 14) demonstrates the ability of the system to change the angle of attack of the kite, spin a shaft in one direction, and produce electricity. The angle of attack system works smoothly, effectively sliding and forcing the model kite to change angle of attack. The shaft system effectively models the sprag-clutch system of the full scale demonstrator, as it allows constant motion of the shafts in one direction. The electrical generation system has a maximum measured voltage output of 1.18 volts. This amount is not enough to effectively power even a small light bulb, but it can be easily read by a voltmeter. This allows us to showcase the system's energy generation capacity.

5.2 Virtual Animation

The final virtual animation is six individual animations combined into one video. It has been made so that it will cycle between all the virtual animations. Each animation is named and labeled where appropriate. The animation has been uploaded to YouTube, so anyone who visits our wiki site (http://www2.me.wpi.edu/wpi-kites) can view it. Table 6 contains information on each segment of the final virtual animation, and explains why it was chosen to be included in the virtual animation.

Table 6: Segments of Final Virtual Animation

Name	What it shows
A-Frame and Arm Motion	A close up view of the A-Frame and arm while the mechanism is in motion
Kite Path	A close up view of the kite while the mechanism is in motion
Instantaneous Power	A zoomed out view of the entire mechanism, alongside a bar graph of the instantaneous power generated according to the MATLAB simulation
Tether Tension	A zoomed out view of the entire mechanism, alongside a bar graph of the tether tension according to the MATLAB simulation
Lift / Drag Ratio	A zoomed out view of the entire mechanism, alongside a bar graph of the lift drag forces on the kite according to the MATLAB simulation
Power Mechanism	A simplified mechanical representation of the mechanism that generates power

5.3 Electrical System

The electrical system we developed fulfills the educational needs of the general public for learning about how a low-cost wind power system can provide much needed electricity and in turn electrical services such as lights. The system we developed, as seen in Figure 29, provides all the functionality that is required to provide a demonstration of different uses of electricity. The electrical components were mounted on a 2-piece hinged plywood display that can stand freely and be easily transported. A wooden carrying holster was created to aid in the transportation of the batteries (Figure 31). The system can successfully charge and store electricity generated from the full sized kite power demonstrator. It can create light and run electronics because it can produce AC electricity. The system can do these things and still prevent damage to the system and its users by its safety features. The electrical system has been successfully implemented to provide a useful educational tool for the public.

6. Future Work

6.1 Scale Model Replica

- Design a wind powered model that does not require A-frame to stabilize the parafoil.
- Channel the airflow of the fan for a more constant and focused flow.
- Improve the oscillation of the arm in the wind powered replica.
- Improve the electric generation capacity of the scale model replica.
- Add an electrical system to the model to further demonstrate electrical generation.
- Custom make the power train components to reduce unwanted vibrations and failure.

6.1.2 Virtual Animation

- Extend the Solid Works model to include a finite element analysis on the A-frame and arm.
- Fully model the power train and angle of attack mechanism and include in the animation.

6.1.3 Electrical System

- Refine system to help reduce costs.
- Optimize system to allow for better manufacturability.

7. References

Beauvais, Joshua, Julia Lyons, Jill Zemianek. Heifer Project International Aquaculture Project. Worcester: Worcester Polytechnic Institute, 2000.

Blouin, Michael, Benjamin Isabella, Joshua Rodden. Wind Power From Kites. Worcester: Worcester Polytechnic Institute, 2007.

Caro, Robert. The Years of Lyndon Johnson: The Path to Power. New York: Alfred a Knopf Inc., 1982.

"Definition of Black Lung Disease." Medicine Net. 2/25/2004. WebMD Network. 16 Apr 2008 <http://www.medterms.com/script/main/art.asp?articlekey=9818>.

"Environmental Concerns." Magenn Power Inc. 27 Aug 2007. Magenn Power Inc. 15 Feb 2008 <http://www.magenn.com/environmentalConcerns.php>.

F.D.J. Nieuwenhout, et al. "Monitoring and evaluation of solar home systems." No. ECN-C-00-089, 2000.

Frey, Jason, et al. Energy Profiling for Off-Grid Energization Solutions in Namibia. Worcester: Worcester Polytechnic Institute, 2007.

Furey, Allister. " Kite Energy." Kite Wind Energy Research. 2007. 15 Feb 2008 <http://www.afurey.com/>.

Goela, J.S., and David J. Olinger "Performance Characteristics of a One-Killowatt Scale Kite Power System." 2008.

Goela, J.S. "Wind Energy Conversion Through Kites." January 1983. Indian Institute of Technology Kanpur.

"Kite For Sail - advanced wind propulsion technology." Kite For Sail. 2007. Kite For Sail, LLC. 15 Apr 2008 <http://kiteforsail.com/index.htm>.

"Kite Gen." Kite Gen. 2007. Kite Gen. 15 Feb 2008 <http://www.kitegen.com/pages/aboutus.html>.

Kruger, Paul. ALTERNATIVE ENERGY RESOURCES: The Quest for Sustainable Energy. 1st. New Jersey: John Wiley & Sons, Inc., 2006.

Lang, David. "Electrical Power Generation Using Kites." Drachen Foundation. December 2005

Lauber, Volkmar. Switching to Renewable Power. 1. Sterling: Earthscan, 2005.

Layton, Julia. "How Wind Power Works." 9 August 2006. HowStuffWorks.com. <http://science.howstuffworks.com/wind-power.htm> 11 April 2008.

Lim, Louisa. "China's Coal-Fueled Boom Has Costs." npr. 2 May 2007 . NPR. 16 Apr 2008 <http://www.npr.org/templates/story/story.php?storyId=9947668>.

Luque, A., Hegedus, S., & Knovel. Handbook of Photovoltaic Science and Engineering. Hoboken: Wiley, 2003. 20 Apr. 2008 <http://www.knovel.com/knovel2/ Toc.jsp?BookID=1081>.

Martinelli, Nicole. "Generating Power From Kites." Wired 10 10 2006 15 Feb 2008 <http://www.wired.com/science/discoveries/news/2006/10/71908>.

Ockels, Wubbo. "Introduction." Laddermill. 2007. Laddermill. 15 Apr 2008 <http://www.ockels.nl/>.

"Oil price hovers near record high." BBC News 28 02 2008 10 03 2008 <http://news.bbc.co.uk/2/hi/business/7270392.stm>.

Olinger, David J., "EPA Proposal," Worcester: Worcester Polytechnic Institute, 2008.

Raymer, Daniel. Aircraft Design: A Conceptual Approach. 3. Sylmar, CA: American Institute of Aeronautics and Astronautics, Inc., 1999.

"Save the Loon with Wind Energy: Comparative Impacts of Wind and Other Energy Sources on Wildlife." American Wind Energy Association. 2008. American Wind Energy Association. 15 Feb 2008 <http://www.awea.org/pubs/factsheets/wildlife.pdf>.

Seccombe, Mike. "Two Sides Debate Cape Wind Plan." 25 September 2007. VOA News. <http://www.mvgazette.com/article.php?912> 11 April 2008.

Tester, Jefferson W., Elisabeth M. Drake, Michael J. Driscoll, Michael W. Golay, and William A. Peters. Sustainable Energy: Choosing Among Options. 1st. Cambridge: MIT Press, 2005.

WHO, UNICEF, Water for Life: Making it Happen. World Heath Organization and UNICEF, 2005.

Wong, Nai Y. et al. Designing and Installing an Alternative Energy System for Heifer Project International. Worcester: Worcester Polytechnic Institute, 2000.

Appendix A: MATLAB Code

```
%  program kiterotarm.m                                April07RUN1
      global g Wk  FLK FDK phi IAD  LL L10 FT  FC  RC  WBA  RA  WDB
WCTR RCTR  RD  W2  R2  IFODE  RG  RG2 RFG  WLOADODE  RF Mac
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t0=0;               %%% define the initial time

g = 9.81;           %%%  grav. acc.  (m/s2)
rho = 1.21;        %%%  air density (kg/m3)
rhofly = 7500.;    %%%  flywheel density (kg/m3)
%%V = 8.5;           %%% ambient wind speed (m/s),  removed when boundary
layer logic added, wind speed set as V0
Ak = 10. ;         %%% wing area  (m2)
L10 = 26.9 ;       %%% tether length (m)
%%%Wk = 0.3+0.1*Ak ; %%% correlation for kite weight (kg)
%%%Wk = Wk*9.81 ;    %%% kite weight in Newtons

Wk = 22.;     %%% kite weight in Newtons

ascent = 1.  ;     %%% ascent/descent flag, changes during run 0-1-0,
etc.
fixedldflag = 0.;    %%% fixed L/D flag
armeffectflag = 0. %%% flag to account for arm rotation effect on AOA
aoashiftgammaflag = 1.;  %%% flag to set AOA shift condition;  1 -
shift AOA at fixed gamma, 0 - shift AOA when VA (arm velocity) = 0.
                            %%% 2 - shift AOA at fixed gamma, reset VA
to
                            %%% zero to model stop
boundaryflag= 1.  ;     %%% boundary layer calculation flag;   1 = 1/7th
power law used,  0 = constant wind speed with elevation
velocityoscflag = 0. ;  %%% wind velocity oscillation flag;  1 =  wind
velocity oscillation at frequency (omegawind  -  Hz), ampwind in
percentage variation about mean wind; phasewind in degrees
                            %%%  where
                            %%%  V0=V0base
                            %%%
+V0base*(ampwind/100.)*sin(omegawind*2*pi*
                            %%%  t1  +phasewind
 flyflag = 0.;       %%%  flywheel engage flag;     1 = flywheel + gear
engaged,     0 =  flywheel + gear NOT engaged
 loopflag = 0.;     %%%  counter for power calculation logic, counts
number of cycles that flywheel is engaged, power calculation only
                     %%%   done for loopglag = 1.



gammaeq = 0.;      %%% equilibrium gamma where spring and counterweight
forces balance
gammastart = -25.;  %%% initial gamma (arm angle) at time t = 0
gammaspring = -70.;  %%% arm angle at which spring force = 0 (deg)
gammaspring = gammaspring*2.*pi/360.;
gammastart = gammastart*2.*pi/360.;
gammaeq = gammaeq*2.*pi/360.;
x03old = gammastart;
```

```
gammamin = -25.;        %%%  arm angle (lower) where AOA is changed
gammamax = 25.;          %%% arm angle (upper) where AOA is changed
gammamin = gammamin*2.*pi/360.;
gammamax = gammamax*2.*pi/360.;

tf=80.;             %%% define the final time
tstart = 23.;  %time start for power calculation to eliminate initial
transient
timepower= 0.;
timepower12= 0.;
timepower21= 0.;
timepowertrans = 0.;
zloadtrans=0.;
vcount = 0.;

nsteps =8000;  nip=4;
eps = 2.0; %%%   rotating arm gamma window (deg) when AOA is changed
(between  + eps and -eps value around gammamin, gammamax)
epsva = 0.25  %%%  rotating arm VA limit (m/s) when AOA is changed
(between + epsva and -epsva value)  same as above line
eps = eps*2.*pi/360.;
epsomega = 0.1;
epsvelstartend = 0.1;

LL = 4.27;   %%% total arm length (meters)
RA = 2.44;   %%%  arm length kite side (meters)
RD = LL-RA;    %%% arm length counterweight side opposite kite(meters)
RC = 2.44;    %%% location of retraction spring and chain from pivot
(meters)
R2 = 2.44;   %%% location of AOA weight mechanism
DELRAOA = 0.5;  %%% PULLEY TO LEADING EDGE KITE CONNECTION DISTANCE ON
ARM (METERS)

Wprime = 65.;  %%% arm weight per unit length  (Newtons/meter)
WDA = Wprime*LL;  %%% total arm weight (Newtons)
WBA = WDA*RA/LL;  %%% arm weight kite side (Newtons)
WDB = WDA*RD/LL;  %%% arm weight counterweight side (Newtons)
W2 = 60.;   %%% AOA weight magnitude + AOA counter weight magnitude
(Newtons)

WCTR = 300.;  %%% counter weight (Newtons),
RCTR = -2.5; %%% radius of counter weight, RCTR > 0 puts weight
opposite kite, RCTR < 0 puts weight on kite side of arm
IAD = 1./12.*WDA/g*LL^2 +  2*W2/g*R2^2  + WCTR/g*RCTR^2 +WDA/g*(LL/2.-
RD)^2 %%% Moment of Inertia of arm::: 1 Rod, 2 AOA weights, 3 Counter
weight  4 Parallel axis theorem

if RCTR > 0.
    kk = WCTR*RCTR/( (gammaeq-gammaspring)*RC^2) %%% spring constant
determined to balance counterweight at equilibrium gamma
else
end
```

```
   kk = 10.   %%% sets spring constant, overrides equilibrium
calculation

slope0 = 2.*pi; %%%  2D airfoil slope
AR = 4.0 ;   %%% wing aspect ratio

kitespan = (AR*Ak)^0.5  %%%kite tip-tip span (meters)
kitechord = kitespan/AR  %%% kite chord   (meters)


alphagasc = 20. ;         %%%  wing geometric angle of attack (ascent)
alphagdsc = -20.   ;       %%% wing geometric angle of attack (descent)
alpha0 = -4.  ;           %%% wing angle of zero lift
alpha0 = alpha0*2.*pi/360.;  %%% convert angle of zero lift to radians
CD0 = 0.1  ;            %%%  parasitic drag coefficient
osweff = 0.9 ;           %%%  span efficiency factor (=1 elliptic)
cmac = 0. ;     %%% wing moment coefficient about aero. center




RG = 0.05;           %%%  gear radius (meters)
RG2 = 0.05;         %%% gear 2 radius (meters)
RFG = 0.025;          %%%  flywheel gear radius (meters)
RF = 0.2;           %%%  radius of flywheel (meters)


TF = .050;          %%% thickness of flywheel (meters)
WF = pi*RF^2*TF*rhofly*g          %%% weight of flywheel (Newtons)
IF = 0.5*WF/g*RF^2   %%%   moment of inertia,  flywheel  (kg - m^2)
WLOAD =80.;       %%% generator load weight  (Newtons)

omegaf = 0.;       %%%  initial flywheel omega  (rad/sec)
omegag = 0.;


slope= slope0/(1. + slope0/pi/AR);

delt = (tf-t0)/nsteps;
deltaoa = 0.25; % time interval over which AOA is changed from
alphagasc0.5 to alphagdsc
delaoa = (alphagasc-alphagdsc)*delt/deltaoa;

alpha = alphagasc ;


theta0 = 70.  ;   %%% initial tether angle with horizontal (deg)
theta0 = theta0*2.0*pi/360.0;

xstart = RA*cos(gammastart)+L10*cos(theta0)
ystart = RA*sin(gammastart)+L10*sin(theta0)
```

```
xstartrel = L10*cos(theta0);
ystartrel = L10*sin(theta0);


x0(1)=0.01;    %%% define x1(0)   VA  arm tip velocity
x0(2)=0.01;      %%% define x2(0)   V2A  kite velocity, normal to
tether w.r.t. A (arm tip)
x0(3)=gammastart;      %%% define x3(0)   gamma (angle of arm)
x0(4)=theta0;   %%% define x4(0)   theta   (angle of tether w.r.t.
horizontal)
xsol(1,:)=x0;
time(1)=t0;
x01old = x0(1);

powerin = 0.;
powerout = 0.;
powerkite12 = 0.;
powerkite21 = 0.;
powerkitetrans12= 0.;
powerkitetrans21= 0.;
powerout12 = 0.;
powerout21 = 0.;
zload12 = 0.;
zload21 = 0.;
zkite12= 0.;
zkite21= 0.;
zctr12 = 0.;
zctr21 = 0.;
zloadtrans12 = 0.;
zloadtrans21 = 0.;

newloadkin12 =0.;
newloadkin21 =0.;
newctrkin12 =0.;
newctrkin21 =0.;
newflykin12 =0.;
newflykin21 =0.;
newkitekin12 = 0.;
newkitekin21 = 0.;
omegafcount = 0.;
omegafcounttrans = 0.;
omegafcountcheck = 0.;


xictr12 = 0.;

powerload = 0.;
powerfly=0.;
zload = 0.;
 zgear = 0.;
 zctr = 0.;
```

```
disty=0.;
V0base = 10.;   %% wind speed at yy0 elevation
omegawind = 0.25;   %%%  wind velocity oscillation frequency in Hz
ampwind =  5.;    %%%  wind velocity oscillation amplitude in % of mean
wind V0
phasewind =  180.;    %%%  phase of wind velocity oscillation (degrees)
wrt arm motion (initial angle).
phasewind = phasewind*2.*pi/360.

yy0 = 10;    %%  reference elevation, elevation of wind speed
measurement
rekite = V0base*100.*kitechord*100./0.15  %%% kite Reynolds number
based on chord
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% closed loop simulation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



   options = odeset('JConstant','on', 'RelTol',1e-6, 'AbsTol',1e-6);
for i=1:nsteps-1

    omegafcountcheck = omegafcountcheck + omegaf*delt;

     t1= t0 + (tf-t0)*(i-1)/nsteps;
    t2= t0 + (tf-t0)*i/nsteps;
    tspan = t1:(t2-t1)/nip:t2;

 if t1 <= tstart + epsvelstartend & t1 >= tstart - epsvelstartend
     velstart = RF*omegaf;
     ctrhstart = RA*x0(3);
 else
 end

  if t1 <= tf + epsvelstartend & t1 >= tf - epsvelstartend
     velend = RF*omegaf;
     ctrhend = RA*x0(3);
 else
 end

    if velocityoscflag == 1.
      V0 =
V0base+V0base*(ampwind/100.)*sin(omegawind*2.*pi*t1+phasewind);
      V = V0;
    else
        V0 = V0base;
        V=V0;
    end


    if boundaryflag == 1.
    yy = disty + ystart;
    V = V0*(yy/yy0)^(1./7.);
```

```
    else
        V = V0;
    end

    vcount = vcount + V *delt;



    Vplot(i+1) = V;
    %disp([ x0(3)   x0(1) ascent ])

    if aoashiftgammaflag == 0.

    if  x0(3) < 0. & x0(1) < epsva  & ascent == 0.
        %disp([ x0(3)   x0(1) ascent ])
        %x0(1) = 0.01;
        ascent = 1.  ;
    else
        end

        if  x0(3) > 0. & x0(1) > -epsva & ascent == 1.
            %disp([ x0(3)   x0(1) ascent Cl])
            %x0(1)  = 0.01;
            ascent = 0.  ;
        else
            end

    else
    end


     if aoashiftgammaflag == 1.

    if  x0(3) < gammamin + eps & x0(3) > gammamin - eps  & ascent == 0.
        disp([ x0(3)   x0(1) ascent   Cl])
        %x0(1) = 0.01;
        ascent = 1.  ;
    else
        end

        if  x0(3) > gammamax - eps & x0(3) < gammamax + eps & ascent ==
1.
            disp([ x0(3)   x0(1) ascent  Cl ])
            %x0(1)  = 0.01;
            ascent = 0.  ;
        else
            end

     else
     end

      if aoashiftgammaflag == 2.

    if  x0(3) < gammamin + eps & x0(3) > gammamin - eps  & ascent == 0.
```

```
        %disp([ x0(3)   x0(1) ascent ])
        x0(1) = 0.01;
        ascent = 1.  ;
    else
        end

        if  x0(3) > gammamax - eps & x0(3) < gammamax + eps & ascent ==
1.
            %disp([ x0(3)   x0(1) ascent ])
            x0(1) = 0.01;
            ascent = 0.  ;
        else
            end

     else
      end

      %%%   calculating top and bottom position times and period of
      %%%   oscillation
      if   x0(1) < epsva & x0(1) > -epsva
        %disp([ i   x0(1) ascent  ])
    else
        end
```

```
    V2 = x0(1)*cos(x0(4)-x0(3)) + x0(2);  %%% check sign on last term
    V1 = -1.*x0(1)*sin(x0(4)-x0(3));

    beta = x0(4)-atan2(V2,V1);
    betaplot(i+1) = beta*360/2./3.14159;
    thetaplot(i+1) = x0(4)*360./2./3.14159;
    gammaplot(i+1) = x0(3)*360./2./3.14159;
    Vk = sqrt( V1^2 + V2^2 );
    Vr = sqrt(  V^2+Vk^2 - 2.*V*Vk*cos(beta) );
    Vkplot(i+1) = Vk;
    Vrplot(i+1) = Vr;
    phi = atan2(Vk*sin(beta),(V-Vk*cos(beta)) );
    phiplot(i+1) = phi*360/2./pi;

    if ascent >= 1.
```

```
        if alpha >= alphagdsc - eps & alpha < alphagasc
            alpha = alpha + delaoa;

        else
        end


        if armeffectflag == 1.
        AOAchange = atan2(DELRAOA*sin(x0(3)),kitechord);
    else
        AOAchange = 0.;
        end


    alphaeff = alpha*2.*pi/360. - phi + AOAchange;

    %alphaeff = 10.*2*pi/360.;  % overrides alpha logic to optimize
alphaeff

    alphaeffplot(i+1)= 360./2./pi*alphaeff;
     alphaplot(i+1) = alpha;

    else
    end

    if ascent == 0.

        if alpha > alphagdsc & alpha <= alphagasc + eps

            alpha = alpha - delaoa;
        else
        end

        if armeffectflag == 1.
        AOAchange = atan2(DELRAOA*sin(x0(3)),kitechord);
    else
        AOAchange = 0.;
        end


    alphaeff = alpha*2.*pi/360. - phi+AOAchange;

    %alphaeff = -4.*2*pi/360.;  % overrides alpha logic to optimize
alphaeff

    alphaeffplot(i+1)= 360./2./pi*alphaeff;
    alphaplot(i+1) = alpha;

    else
    end
```

```
    Cl = slope*(alphaeff - alpha0);

    Clplot(i+1)= Cl;
    Cd = CD0 +  Cl^2/(pi*AR*osweff);
    Cdplot(i+1) = Cd;
     LD=Cl/Cd;
     LDplot(i+1) = LD;


   %  if fixedldflag == 1. & ascent == 1.
    %     LD = 6.5;
     %    Cl = 1.;
      %   Cd = Cl/LD;
      %   Clplot(i+1)= Cl;
      %   Cdplot(i+1) = Cd;
      %   LDplot(i+1) = LD;
    % else
    %     end

   %  if fixedldflag == 1. & ascent == 0.
    %     LD = 0.65;
     %    Cl = 0.1;
    %     Cd = Cl/LD;
     %    Clplot(i+1)= Cl;
       % Cdplot(i+1) = Cd;
      %  LDplot(i+1) = LD;
     %else
      %  end



    FLK = 0.5*rho*(Vr^2)*Cl*Ak;
    FDK = 0.5*rho*(Vr^2)*Cd*Ak;

    Mac = cmac*0.5*rho*(Vr^2)*Ak*kitechord ;  %%%  moment about aero.
center for wing

    FT = FDK*cos(x0(4)+phi)+FLK*sin(x0(4)+phi)-
Wk*sin(x0(4))+Wk*x0(2)^2/g/L10;
     FC = (x0(3)-gammaspring)*RC*kk;

   %%%FT = 0.;


     if FT <= 0
         FT=0.;
    else
     end
```

```
     FTplot(i+1) = FT;
     FCplot(i+1) = FC;




    %disp([ i   x0(3) FC    FT  Cl  ])


    x03old = x0(3);
    x01old = x0(1);


    %%%  Flywheel logic

    omega = -x0(1)/RA;
    omegag = omega*RC/RG;


    omegagplot(i+1) = omegag;

    zgear = zgear +RG*omegag*delt;
    zgearplot(i+1) = zgear;
    omegafstar = omegaf*RFG/RG2;
    if omegag >= omegafstar

        IFODE = IF;
        WLOADODE = WLOAD;       %%%  If WLOADODE = 0.  load NOT attached
during engagement,  If WLOADODE = WLOAD  load  ATTACHED during
engagement
         omegaf = omegag*RG2/RFG;
        omegafplot(i+1) = omegaf;

    else
        IFODE = 0.;
        WLOADODE = 0.;
        omegaf = omegaf-WLOAD*RF/IF*delt;
        omegafplot(i+1) = omegaf;
    end

   %disp ([t1 omegaf])


    [t,x] = ode23s('xprimekiterotarmdec162007m',tspan,x0,options);
    x0=x(nip+1,:);
    xsol(i+1,:) = x0;
    time(i+1)=t2;

    %%%  Power calculation logic

    TorqueFT = FT*RA*cos(x0(3)+pi/2.-x0(4));
    TorqueFC = FC*RC*cos(x0(3));
```

```
    powerFT = TorqueFT*omega;
    powerFC = TorqueFC*omega;

    %powerplot(i+1) = power;
    %Torqueplot(i+1) = Torque;
    TorqueFTplot(i+1) = TorqueFT;
    TorqueFCplot(i+1) = TorqueFC;
    omegaplot(i+1) = omega;
    powerFTplot(i+1) = powerFT;
    powerFCplot(i+1) = powerFC;

    %omegafrun = omegafcounttrans/timepowertrans;
     %disp([t1 powerkitetrans omegaf   flyflag  ])




    if t1 == tstart

    disp ([t1 flyflag omegag omegaf])
    else
    end

    if  t1 < tstart & omegag >= omegafstar

        powerkitetrans12 = powerkitetrans12 + FT*Vk*cos(x0(4)-
beta)*delt;
        timepowertrans = timepowertrans + delt;
        omegafcounttrans = omegafcounttrans + omegaf*delt;
        zloadtrans12 = zloadtrans12 + RF*omegaf*delt;
    else
    end

    if  t1 < tstart & omegag < omegafstar

        powerkitetrans21 = powerkitetrans21 + FT*Vk*cos(x0(4)-
beta)*delt;
       zloadtrans21 = zloadtrans21 + RF*omegaf*delt;

    else
    end




    zload21 = zload21 + 0.;
    zload21nickplot(i+1)= zload21;
    zload12nickplot(i+1)= zload12;




    zctr12 = zctr12 + 0.;
    zctr12plot(i+1) = zctr12;
```

```
      zload21plot(i+1) = zload21*WLOAD;
      zload12plot(i+1) = zload12*WLOAD;
      instloadpowplot(i+1) = 0.;
    instkitepowplot(i+1) = 0.;
    powerkite12 = powerkite12 + 0.;
    powerkite12plot(i+1) = powerkite12;

    if t1 > tstart


  if   omegag >= omegafstar
      %%disp ([flyflag  loopflag])
      if flyflag == 0.
          loopflag = loopflag + 1.




      flyflag = 1.;
      else
      end




     if loopflag >= 1. &  loopflag <=400.

         omegafcount = omegafcount + omegaf*delt;
      powerout12 = powerout12 + powerFT*delt;     %%% power at pivot


    %powerkite12 = powerkite12 + FT*Vk*cos(x0(4) - beta )*delt;  %%%
power from kite
    acckite = Wk/g*(Vkplot(i+1)-Vkplot(i))/delt;
    %disp ([acckite  FT])
    powerkite12 = powerkite12 + (FT+Wk/g*(Vkplot(i+1)-
Vkplot(i))/delt)*Vk*cos(x0(4) - beta )*delt;  %%% power from kite
    powerkite12plot(i+1) = powerkite12;


     %powerload = powerload + FT*RA/RF*RG/RC*cos(x0(3)+pi/2.-
x0(4))*RF*omegaf*delt;
      zload12 = zload12 + RF*omegaf*delt;
      zload12plot(i+1) = zload12*WLOAD;
      zload12nickplot(i+1) = zload12;
      zctr12 = zctr12 + RA*omega*delt;
      zkite12 = zkite12 + Vk*sin(beta)*delt;


      zctr12plot(i+1) = zctr12;




     %disp ([t1 loopflag  flyflag   zctr12 zctr21 powerkite12
powerkite21])
```

```
     %disp ([t1 loopflag  flyflag   omegaf omegafstar])

      newctrkin12 = newctrkin12 + 0.5*WCTR/g*RA^2*(omegaplot(i+1)^2-
omegaplot(i)^2);
          newloadkin12 = newloadkin12 +
0.5*WLOAD/g*RF^2*(omegafplot(i+1)^2-omegafplot(i)^2);
           newflykin12 = newflykin12 + 0.5*IF*(omegafplot(i+1)^2-
omegafplot(i)^2);
           newkitekin12 = newkitekin12 + 0.5*Wk/g*(Vkplot(i+1)^2-
Vkplot(i)^2);


       timepower12 = timepower12 + delt;
      timepower = timepower + delt;


    %instkitepow = FT*Vk*cos(x0(4)-beta);
    instloadpow = (zload12plot(i+1)- zload12plot(i))/delt;
    instloadpowplot(i+1) = instloadpow;
    %instkitepowplot(i+1) = instkitepow;
    %disp([instkitepow  instloadpow])
     else
    end


  else
      flyflag = 0.;
      %%disp ([flyflag  loopflag])

     if loopflag >= 1. &  loopflag <=400.




         omegafcount = omegafcount + omegaf*delt;
      zload21 = zload21 + RF*omegaf*delt;
      zload21plot(i+1) = zload21*WLOAD;
      zload21nickplot(i+1) = zload21;



      zctr21 = zctr21 + RA*omega*delt;
      zkite21 = zkite21 + Vk*sin(beta)*delt;

        %powerkite21 = powerkite21 + FT*Vk*cos(x0(4)-beta)*delt;
        powerkite21 = powerkite21 + (FT+Wk/g*(Vkplot(i+1)-
Vkplot(i))/delt)*Vk*cos(x0(4) - beta )*delt;  %%% power from kite

      % disp ([t1 loopflag  flyflag   zctr12 zctr21 powerkite12
powerkite21])
       %disp ([t1 loopflag  flyflag   omegaf omegafstar])
        powerout21 = powerout21 + powerFT*delt;
```

```
        newctrkin21 = newctrkin21 + 0.5*WCTR/g*RA^2*(omegaplot(i+1)^2-
omegaplot(i)^2);
           newloadkin21 = newloadkin21 +
0.5*WLOAD/g*RF^2*(omegafplot(i+1)^2-omegafplot(i)^2);
           newflykin21 = newflykin21 + 0.5*IF*(omegafplot(i+1)^2-
omegafplot(i)^2);
           newkitekin21 = newkitekin21 + 0.5*Wk/g*(Vkplot(i+1)^2-
Vkplot(i)^2);

           %instkitepow = FT*Vk*cos(x0(4)-beta);
          % instkitepowplot(i+1) = instkitepow;
          instloadpow = (zload21plot(i+1)- zload21plot(i))/delt;
    instloadpowplot(i+1) = instloadpow;

      timepower = timepower + delt;

       timepower21 = timepower21 + delt;
        else
        end
  end

    else



    %  turning on next 6 lines calculates instant. kite power during
    %  initial transient
   %  instkitepow = FT*Vk*cos(x0(4)-beta);
  %  instloadpow = WLOAD*(zload12plot(i+1)- zload12plot(i))/delt;
   % instloadpowplot(i+1) = instloadpow;
   % instkitepowplot(i+1) = instkitepow;
    %instloadpow = WLOAD*(zload21plot(i+1)- zload21plot(i))/delt;
    %instloadpowplot(i+1) = instloadpow;

    end


    poweroutplot(i+1) = powerout;
    powerinplot(i+1) = powerin;
     %powerkiteplot(i+1) = powerkite;


    distx = RA*cos(x0(3))+ L10*cos(x0(4))-xstart;
    disty = RA*sin(x0(3))+ L10*sin(x0(4))-ystart;

    distxrela = L10*cos(x0(4))-xstartrel;
    distyrela = L10*sin(x0(4))-ystartrel;

    distxplot(i+1) = distx;
    distyplot(i+1) = disty;
```

```
    distxrelaplot(i+1) = distxrela;
    distyrelaplot(i+1) = distyrela;

    %thetaplot(i+1) = x0(4)*360/2./3.14159;
    pbeta = beta*360/2./pi;
    pphi = phi*360./2./pi;
    palphaeff = alphaeff*360./2./pi;
    %disp([t1  x0(1) x0(2) x0(3) x0(4)])
    %disp([t1 ascent alpha palphaeff  ])



end

            newloadpot12 = (WLOAD*zload12)/timepower
            newloadpot21 = (WLOAD*zload21)/timepower
            newloadkin12 = newloadkin12/timepower
            newloadkin21 = newloadkin21/timepower



            newctrpot12 = (WCTR*zctr12)/timepower
            newctrpot21 = (WCTR*zctr21)/timepower
            newctrkin12 = newctrkin12/timepower
            newctrkin21 = newctrkin21/timepower


            newflykin12 = newflykin12/timepower
            newflykin21 = newflykin21/timepower

            newkitepot12 = (Wk*zkite12)/timepower
            newkitepot21 = (Wk*zkite21)/timepower
            newkitekin12 = newkitekin12/timepower
            newkitekin21 = newkitekin21/timepower

            powerkite12 = powerkite12/timepower
            powerkite21 = powerkite21/timepower
            powerout12 = powerout12/timepower
            powerout21 = powerout21/timepower;

             newloadtrans12 = (WLOAD*zloadtrans12)/timepower
              newloadtrans21 = (WLOAD*zloadtrans21)/timepower

            powerout = powerout12 + powerout21;
            newloadpot = newloadpot12  + newloadpot21;



            omegaavg = omegafcount/timepower
            poweromegaavg = RF*omegaavg*WLOAD
```

```
            omegaavgcheck = omegafcountcheck/timepower
            poweromegaavgcheck = RF*omegaavgcheck*WLOAD


            omegaenergy = 0.5*IF*omegaavg^2
           % disp ([powerkitetrans])

            powerfix = omegafcount/timepower*RF*WLOAD;
            powerfixkite = powerfix + powerkite12;

            powerkitetrans12 = powerkitetrans12/timepower
            powerkitetrans21 = powerkitetrans21/timepower




            disp([powerkite12 powerkite21 ])
            %disp([powerout12 powerout21 powerout])
            disp([ powerfix ])

          disp ([timepower12 timepower21 timepower timepowertrans
newloadkin21])

       % disp ([newctrpot12 newflykin12   newctrkin12 newkitepot12
newkitekin12 powerkite12])
        %disp ([newloadpot21  newloadkin21   newflykin21 ])
        %disp  ([newctrpot21  newctrkin21 newkitepot21 newkitekin21
powerkite21])
        %disp ([newloadpot21  newloadkin21  powerkite]

        check1sys =
newctrpot12+newctrkin12+newloadpot12+newloadkin12+newkitepot12+newkitek
in12+newflykin12
        check1kite = powerkite12+(powerkitetrans12 + powerkitetrans21)
        check1 = check1sys - check1kite

       check2 = newloadpot21   + newflykin21+newloadkin21
      check3 = newctrpot21    + newctrkin21 +newkitepot21 +
newkitekin21  -   powerkite21
     check4 = check1 + check2 + check3

     disp([newloadpot12 newloadpot21 newloadpot])
     vavg = vcount/tf
               cpavg = newloadpot/(0.5*rho*vavg^3*Ak)
               WLOADBAR = WLOAD/(0.5*rho*vavg^2*Ak)

       %check4 =    powerkite12 - powerkite21
       %disp ([timepower zload21])


       % eloadpot = (WLOAD * zload)/timepower
        %ectrpot = (WCTR * zctr)/timepower
        %ectrstart = ectrstart/timepower
        %ectrmid = ectrmid/timepower
```

```
        %eflystart = eflystart/timepower
        %eflymid = eflymid/timepower



    %poweroutavg = powerout/timepower
      %powerkiteavg = powerkite/timepower
           % cpoutavg = poweroutavg/(0.5*rho*V^3*Ak)
     %powercheck = powerload/timepower

   %cpload = powerloadavg/(0.5*rho*V^3*Ak)



   n1 = gammaplot';
    n2 = alphaeffplot';
    n3 = phiplot';
    n4 = distxplot';
    n5 = distyplot';
    n6 = FTplot';
    n7 = FCplot';
    n8 = powerFTplot';
  %  n9 = powerkiteplot';
    n10 = omegaplot';
    n11 = time';
    n12 = Clplot';
    n13 = Cdplot';
    n14 = LDplot';
    n15 = omegafplot';
    n16 = omegaplot';
    n17 = omegagplot';
    n18 = instloadpowplot';
    n19 = betaplot';
    n20 = Vrplot';
    n21 = Vkplot';
    n22 = zload12nickplot';
    n23 = zload21nickplot';
    n24 = thetaplot';
    n25 = alphaplot';


   %save('sept07run1angles.dat','n11','n1','n2','n3')
   %save('sept07run1kitemotion.dat','n11','n4','n5')
   % save('sept07run1forces.dat','n11','n6','n7')
    %save('sept07run1power.dat','n11','n8','n10')
    % save('sept07run1kiteforces.dat','n11','n12','n13','n14')
    % save('feb08run1figure4.dat','n11','n15','n16','n17')
      %save('feb08run1figure5c.dat','n11','n1','n18','n6')


    %% This one is the final case for the first animation
    %%M=horzcat(n11, n18, n6,  n14);
    %% define 42 * 4 matrix for the key values
    %%
N=[0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,
```

```
0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0
,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,
0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0
,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0];

    %% This is for the power mechanism animation
    %% M=horzcat(n11, n15, n16, n17, n22, n23);
    %% define 42 * 6 matrix for the key values
    %%
N=[0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,
0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0
,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,
0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0
,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,
0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0
,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,
0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0;0
,0,0,0,0,0;0,0,0,0,0,0];


    %%Used to check arm motion in power mechanism animation
    M=horzcat(n11, n17, n22, n23, n1);
    %% define 42 * 5 matrix for the key values

N=[0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,
0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0
;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;
0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0
,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,
0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0];



    r= 0;

    for q=3:600
    N(q,:)=M(2+10*r,:);
    r=r+1;
    end

    xlswrite('mar08nick',N);
    %xlswrite('test',N);




subplot(211),plot(time,gammaplot)
legend('gamma(t)',0)
%subplot(511),plot(time,Vplot',':')
%legend('Wind speed',0)
%subplot(411),plot(time,instloadpowplot)
```

```
%legend('Inst Pow',0)
%subplot(312),plot(time,zload12plot,time,zload21plot,time,powerkite12pl
ot)
%legend('Inst Pow',0)
%subplot(612),plot(time,Clplot)
%legend('Lift Coefficient',0)
%subplot(411),plot(time,omegafplot,time,omegaplot,time,omegagplot)
%legend('Omegas' ,0)
%subplot(312),plot(time,zgearplot)
%legend('Zgear' ,0)
%subplot(712),plot(time,TorqueFTplot,time,TorqueFCplot)
%legend('TorqueFT' ,'TorqueFC' ,0)
subplot(212),plot(time,distxplot, time,distyplot)
legend('Kite motion',0)
%subplot(413),plot(time,powerFTplot)
%legend('PowerOut',0)
%subplot(313),plot(time,FTplot,time,FCplot)
%legend('Line Tension','Spring Force',0)
%subplot(514),plot(time,phiplot,time,alphaeffplot,':')
%legend('Phi','Alphaeff' ,0)
%subplot(412),plot(distxplot,distyplot)
%legend('Kite Motion ',0)
%subplot(414),plot(distxrelaplot,distyrelaplot)
%legend('Parafoil Motion - Relative to A',0)
%%%subplot(514),plot(Cdplot,Clplot)
%%%legend('Lift-Drag',0)
%subplot(111),plot(alphaeffplot,Clplot)
%legend('Lift Curve',0)
%subplot(413),plot(time,Vkplot,time,Vrplot,':')
%legend('Vk','Vr',0)
%subplot(414),plot(time,Clplot,time,Cdplot,':')
%legend('Cl','Cd',0)
%subplot(414),plot(time,LDplot)
%legend('L/D',0)
```

Appendix B: List of Sub-Components in Virtual Animation

A-Frame and Kite Animation

- A-Frame
- 3 Piece Rocking Arm
- Pivot Point
    - Rod
    - Pillow blocks (http://www.3dcontentcentral.com/)
    - Plate
- Sleeve
- Tether
- Kite

Power Mechanism Animation

- Sprag Clutch
- Flywheel
- Wall mount
- Spring
- Arm Representation
- Chain
- Weight

Appendix C: Tables Containing Final Case Input Data

Gamma

Time	Gamma (Output from MATLAB)	Gamma (Used in SolidWorks)
0	34.02118323	55.97881677
0.5	45.53602782	44.46397218
1	25.04936773	64.95063227
1.5	-28.3130801	118.3130801
2	-37.91406422	127.9140642
2.5	-3.73433013	93.73433013
3	31.05145544	58.94854456
3.5	45.87574922	44.12425078
4	28.72482542	61.27517458
4.5	-21.6305754	111.6305754
5	-42.82982229	132.8298223
5.5	-9.493015769	99.49301577
6	25.08967505	64.91032495
6.5	45.93961606	44.06038394
7	34.77186253	55.22813747
7.5	-9.647656994	99.64765699
8	-46.88765477	136.8876548
8.5	-14.54923771	104.5492377
9	19.83111372	70.16888628
9.5	44.86077089	45.13922911
10	38.12910351	51.87089649
10.5	-1.679784258	91.67978426
11	-50.56399827	140.5639983
11.5	-19.96262529	109.9626253
12	14.22763769	75.77236231
12.5	43.06080588	46.93919412
13	41.36922382	48.63077618
13.5	6.975448275	83.02455173
14	-50.50959506	140.5095951
14.5	-25.19774678	115.1977468
15	9.147416364	80.85258364
15.5	40.50934168	49.49065832
16	43.08276529	46.91723471
16.5	13.20034419	76.79965581
17	-45.35685614	135.3568561
17.5	-28.67301095	118.673011
18	5.735885587	84.26411441
18.5	38.5173186	51.4826814
19	44.2634463	45.7365537
19.5	17.77887282	72.22112718
20	-39.74513368	129.7451337

Theta

Time	Theta (Output from MATLAB, and Used in SolidWorks)
0	64.09627746
0.5	57.82699848
1	55.27859286
1.5	56.43738485
2	67.38691986
2.5	66.42153835
3	64.51644429
3.5	58.30483507
4	55.34557045
4.5	56.22475874
5	66.82687539
5.5	66.98760664
6	64.77753382
6.5	59.26925745
7	55.5777946
7.5	55.92129842
8	65.44106409
8.5	67.38961695
9	64.83708366
9.5	60.06232701
10	55.83271331
10.5	55.66483937
11	63.64819332
11.5	67.79359461
12	65.07996375
12.5	61.1480524
13	56.28615574
13.5	55.4653491
14	60.73237094
14.5	67.97388554
15	65.32314162
15.5	62.13241999
16	56.71535155
16.5	55.33728548
17	58.70566139
17.5	67.95546946
18	65.58682776
18.5	62.91706532
19	57.08604725
19.5	55.2826117
20	57.55152053

New Alpha

Time	New Alpha (Output from MATLAB)	New Alpha (Used in SolidWorks)
0	-7.2	82.8
0.5	-20	70
1	-20	70
1.5	-12	78
2	20	110
2.5	20	110
3	0.8	90.8
3.5	-20	70
4	-20	70
4.5	-20	70
5	20	110
5.5	20	110
6	15.2	105.2
6.5	-20	70
7	-20	70
7.5	-20	70
8	20	110
8.5	20	110
9	20	110
9.5	-20	70
10	-20	70
10.5	-20	70
11	20	110
11.5	20	110
12	20	110
12.5	-20	70
13	-20	70
13.5	-20	70
14	20	110
14.5	20	110
15	20	110
15.5	-20	70
16	-20	70
16.5	-20	70
17	10.4	100.4
17.5	20	110
18	20	110
18.5	-20	70
19	-20	70
19.5	-20	70
20	2.4	92.4

Power

Time	Power (Output from MATLAB)	Power / 2 (Used in SolidWorks)
0	1964.149261	982.0746307
0.1	1936.986818	968.4934088
0.2	1909.824374	954.912187
0.3	1882.66193	941.3309652
0.4	1855.499487	927.7497434
0.5	1828.337043	914.1685216
0.6	1801.1746	900.5872998
0.7	1774.012156	887.006078
0.8	1746.849712	873.4248562
0.9	1719.687269	859.8436344
1	1692.524825	846.2624125
1.1	1665.362381	832.6811907
1.2	1638.199938	819.0999689
1.3	1611.037494	805.5187471
1.4	1583.875051	791.9375253
1.5	1556.712607	778.3563035
1.6	1529.550163	764.7750817
1.7	1502.38772	751.1938599
1.8	1475.225276	737.6126381
1.9	1495.669205	747.8346024
2	1692.979193	846.4895966
2.1	1860.260578	930.1302891
2.2	1905.587633	952.7938167
2.3	1897.813542	948.9067708
2.4	1873.469047	936.7345235
2.5	1856.073828	928.0369138
2.6	1848.091196	924.0455978
2.7	1868.304468	934.1522342
2.8	1920.626536	960.3132678
2.9	2000.283158	1000.141579
3	1978.377823	989.1889117
3.1	1951.21538	975.6076899
3.2	1924.052936	962.0264681
3.3	1896.890493	948.4452463
3.4	1869.728049	934.8640245
3.5	1842.565605	921.2828027
3.6	1815.403162	907.7015809
3.7	1788.240718	894.1203591
3.8	1761.078275	880.5391373
3.9	1733.915831	866.9579155
4	1706.753387	853.3766936
4.1	1679.590944	839.7954718
4.2	1652.4285	826.21425
4.3	1625.266056	812.6330282
4.4	1598.103613	799.0518064
4.5	1570.941169	785.4705846

Power

Time	Power (Output from MATLAB)	Power / 2 (Used in SolidWorks)
4.6	1543.778726	771.8893628
4.7	1516.616282	758.308141
4.8	1489.453838	744.7269192
4.9	1477.458393	738.7291966
5	1486.077327	743.0386636
5.1	1729.308584	864.6542922
5.2	1868.822393	934.4111966
5.3	1899.847921	949.9239603
5.4	1886.163165	943.0815827
5.5	1861.722398	930.8611989
5.6	1847.587988	923.7939942
5.7	1844.29557	922.1477851
5.8	1870.239414	935.1197069
5.9	1927.866007	963.9330035
6	2002.579182	1001.289591
6.1	1975.416739	987.7083694
6.2	1948.254295	974.1271476
6.3	1921.091852	960.5459258
6.4	1893.929408	946.964704
6.5	1866.766964	933.3834822
6.6	1839.604521	919.8022603
6.7	1812.442077	906.2210385
6.8	1785.279633	892.6398167
6.9	1758.11719	879.0585949
7	1730.954746	865.4773731
7.1	1703.792303	851.8961513
7.2	1676.629859	838.3149295
7.3	1649.467415	824.7337077
7.4	1622.304972	811.1524859
7.5	1595.142528	797.5712641
7.6	1567.980084	783.9900422
7.7	1540.817641	770.4088204
7.8	1513.655197	756.8275986
7.9	1486.492754	743.2463768
8	1488.676747	744.3383735
8.1	1492.144347	746.0721735
8.2	1778.365435	889.1827176
8.3	1888.697863	944.3489316
8.4	1904.556614	952.2783068
8.5	1887.928507	943.9642533
8.6	1863.983325	931.9916627
8.7	1850.640907	925.3204536
8.8	1851.941862	925.970931
8.9	1886.117415	943.0587073
9	1951.020515	975.5102577

Power

Time	Power (Output from MATLAB)	Power / 2 (Used in SolidWorks)
9.1	1996.643677	998.3218383
9.2	1969.481233	984.7406165
9.3	1942.318789	971.1593947
9.4	1915.156346	957.5781729
9.5	1887.993902	943.996951
9.6	1860.831458	930.4157292
9.7	1833.669015	916.8345074
9.8	1806.506571	903.2532856
9.9	1779.344128	889.6720638
10	1752.181684	876.090842
10.1	1725.01924	862.5096202
10.2	1697.856797	848.9283984
10.3	1670.694353	835.3471766
10.4	1643.53191	821.7659548
10.5	1616.369466	808.1847329
10.6	1589.207022	794.6035111
10.7	1562.044579	781.0222893
10.8	1534.882135	767.4410675
10.9	1507.719691	753.8598457
11	1480.557248	740.2786239
11.1	1485.756943	742.8784714
11.2	1561.019358	780.5096788
11.3	1801.619419	900.8097094
11.4	1887.643511	943.8217553
11.5	1896.97904	948.4895199
11.6	1875.975878	937.9879392
11.7	1851.993305	925.9966526
11.8	1843.905086	921.9525432
11.9	1852.905289	926.4526447
12	1894.486695	947.2433476
12.1	1965.429405	982.7147023
12.2	1989.72852	994.8642602
12.3	1962.566077	981.2830384
12.4	1935.403633	967.7018165
12.5	1908.241189	954.1205947
12.6	1881.078746	940.5393729
12.7	1853.916302	926.9581511
12.8	1826.753859	913.3769293
12.9	1799.591415	899.7957075
13	1772.428971	886.2144857
13.1	1745.266528	872.6332639
13.2	1718.104084	859.0520421
13.3	1690.94164	845.4708202
13.4	1663.779197	831.8895984
13.5	1636.616753	818.3083766

Power

Time	Power (Output from MATLAB)	Power / 2 (Used in SolidWorks)
13.6	1609.45431	804.7271548
13.7	1582.291866	791.145933
13.8	1555.129422	777.5647112
13.9	1527.966979	763.9834894
14	1500.804535	750.4022676
14.1	1473.642092	736.8210458
14.2	1492.25115	746.1255751
14.3	1692.78613	846.3930648
14.4	1863.063289	931.5316445
14.5	1909.222905	954.6114526
14.6	1898.034491	949.0172456
14.7	1875.111018	937.5555091
14.8	1854.161176	927.0805882
14.9	1848.890335	924.4451677
15	1869.271615	934.6358077
15.1	1921.991876	960.9959378
15.2	2002.006952	1001.003476
15.3	1980.119075	990.0595373
15.4	1952.956631	976.4783155
15.5	1925.794187	962.8970937
15.6	1898.631744	949.3158719
15.7	1871.4693	935.73465
15.8	1844.306856	922.1534282
15.9	1817.144413	908.5722064
16	1789.981969	894.9909846
16.1	1762.819526	881.4097628
16.2	1735.657082	867.828541
16.3	1708.494638	854.2473192
16.4	1681.332195	840.6660974
16.5	1654.169751	827.0848756
16.6	1627.007307	813.5036537
16.7	1599.844864	799.9224319
16.8	1572.68242	786.3412101
16.9	1545.519977	772.7599883
17	1518.357533	759.1787665
17.1	1491.195089	745.5975447
17.2	1489.750885	744.8754426
17.3	1496.896961	748.4484806
17.4	1780.067054	890.033527
17.5	1889.04127	944.5206349
17.6	1904.870356	952.4351781
17.7	1887.055515	943.5277576
17.8	1865.838031	932.9190153
17.9	1851.064002	925.5320008
18	1851.881471	925.9407357

Power

Time	Power (Output from MATLAB)	Power / 2 (Used in SolidWorks)
18.1	1885.12951	942.5647548
18.2	1949.0681	974.5340498
18.3	1994.104877	997.0524384
18.4	1966.942433	983.4712166
18.5	1939.77999	969.8899948
18.6	1912.617546	956.3087729
18.7	1885.455102	942.7275511
18.8	1858.292659	929.1463293
18.9	1831.130215	915.5651075
19	1803.967771	901.9838857
19.1	1776.805328	888.4026639
19.2	1749.642884	874.8214421
19.3	1722.480441	861.2402203
19.4	1695.317997	847.6589985
19.5	1668.155553	834.0777767
19.6	1640.99311	820.4965548
19.7	1613.830666	806.915333
19.8	1586.668222	793.3341112
19.9	1559.505779	779.7528894
20	1532.343335	766.1716676

Tension

Time	Tension (Output from MATLAB)	Tension / 3 (Used in SolidWorks)
0	0	0
0.1	27.89112001	9.297040005
0.2	52.87782181	17.6259406
0.3	54.01947577	18.00649192
0.4	54.86558443	18.28852814
0.5	56.00026779	18.66675593
0.6	57.91579792	19.30526597
0.7	61.15319548	20.38439849
0.8	66.3056532	22.1018844
0.9	74.00556767	24.66852256
1	84.83686546	28.27895515
1.1	99.0916526	33.03055087
1.2	116.2963642	38.76545472
1.3	134.5751493	44.8583831
1.4	150.295055	50.09835165
1.5	363.826699	121.2755663
1.6	1051.872222	350.6240739
1.7	3494.233755	1164.744585
1.8	1406.161372	468.7204574
1.9	1020.128458	340.0428195
2	724.4543611	241.484787
2.1	489.8912679	163.2970893
2.2	380.0534135	126.6844712
2.3	344.3673007	114.7891002
2.4	355.2676471	118.422549
2.5	400.3300587	133.4433529
2.6	476.8354523	158.9451508
2.7	574.3722966	191.4574322
2.8	688.2290463	229.4096821
2.9	594.0908008	198.0302669
3	0	0
3.1	0	0
3.2	50.1696373	16.72321243
3.3	53.55972071	17.85324024
3.4	54.42471371	18.14157124
3.5	55.34623193	18.44874398
3.6	56.79302085	18.93100695
3.7	59.28116297	19.76038766
3.8	63.37634105	21.12544702
3.9	69.69559544	23.23186515
4	78.86655817	26.28885272
4.1	91.37494646	30.45831549
4.2	107.2136492	35.73788306
4.3	125.3019441	41.76731469
4.4	142.9001499	47.63338331
4.5	155.6761066	51.89203554

Tension

Time	Tension (Output from MATLAB)	Tension / 3 (Used in SolidWorks)
4.6	539.2847134	179.7615711
4.7	1725.61895	575.2063166
4.8	2429.404765	809.8015882
4.9	1166.636768	388.8789227
5	967.3452592	322.4484197
5.1	654.5167054	218.1722351
5.2	452.4814744	150.8271581
5.3	363.5514673	121.1838224
5.4	339.5837238	113.1945746
5.5	359.0026714	119.6675571
5.6	410.4128873	136.8042958
5.7	491.4797677	163.8265892
5.8	591.839957	197.2799857
5.9	707.8460054	235.9486685
6	496.9040361	165.6346787
6.1	0	0
6.2	0	0
6.3	51.23081018	17.07693673
6.4	53.6592985	17.88643283
6.5	54.45523864	18.15174621
6.6	55.34732125	18.44910708
6.7	56.80309525	18.93436508
6.8	59.3387399	19.77957997
6.9	63.52101832	21.17367277
7	69.96942195	23.32314065
7.1	79.31231948	26.43743983
7.2	92.02659381	30.67553127
7.3	108.0731672	36.02438906
7.4	126.3021927	42.1007309
7.5	143.8728539	47.95761798
7.6	156.3691954	52.12306513
7.7	599.4222899	199.80743
7.8	1964.891866	654.9639554
7.9	2210.31177	736.7705902
8	1131.447138	377.149046
8.1	934.6961836	311.5653945
8.2	600.4708687	200.1569562
8.3	426.0317803	142.0105934
8.4	354.8411649	118.2803883
8.5	341.1332018	113.7110673
8.6	369.0004121	123.0001374
8.7	428.7392173	142.9130724
8.8	515.9558585	171.9852862
8.9	621.0828531	207.0276177
9	742.5145601	247.5048534

Tension

Time	Tension (Output from MATLAB)	Tension / 3 (Used in SolidWorks)
9.1	211.1643195	70.3881065
9.2	0	0
9.3	13.43270921	4.477569736
9.4	52.66349932	17.55449977
9.5	53.91826424	17.97275475
9.6	54.71347468	18.23782489
9.7	55.73962732	18.57987577
9.8	57.4801689	19.1600563
9.9	60.46714042	20.15571347
10	65.28495068	21.76165023
10.1	72.56401755	24.18800585
10.2	82.90931657	27.63643886
10.3	96.68575435	32.22858478
10.4	113.5782483	37.8594161
10.5	131.9573543	43.98578476
10.6	148.4289493	49.47631645
10.7	304.181502	101.393834
10.8	832.3027871	277.4342624
10.9	2830.865102	943.6217006
11	1695.736854	565.2456179
11.1	1075.499047	358.4996822
11.2	846.1737874	282.0579291
11.3	554.2336932	184.7445644
11.4	407.0753041	135.691768
11.5	349.1549994	116.3849998
11.6	347.2236202	115.7412067
11.7	381.109757	127.0365857
11.8	447.0204579	149.0068193
11.9	537.7330261	179.244342
12	645.3679622	215.1226541
12.1	770.2563965	256.7521322
12.2	71.73916341	23.91305447
12.3	0	0
12.4	36.85731566	12.28577189
12.5	53.07658184	17.69219395
12.6	54.07673229	18.02557743
12.7	54.8818112	18.29393707
12.8	56.01214449	18.67071483
12.9	57.96085181	19.32028394
13	61.2695582	20.42318607
13.1	66.53348786	22.17782929
13.2	74.38703301	24.79567767
13.3	85.41063629	28.4702121
13.4	99.87654384	33.29218128
13.5	117.2600198	39.08667325

Tension

Time	Tension (Output from MATLAB)	Tension / 3 (Used in SolidWorks)
13.6	135.5959961	45.19866535
13.7	151.1514465	50.38381548
13.8	365.0705687	121.6901896
13.9	1053.512962	351.1709873
14	3503.394229	1167.798076
14.1	1419.929671	473.3098905
14.2	1030.607456	343.5358188
14.3	730.5431306	243.5143769
14.4	491.0682848	163.6894283
14.5	378.0759449	126.025315
14.6	340.4412997	113.4804332
14.7	349.0766025	116.3588675
14.8	393.4407611	131.1469204
14.9	466.4227583	155.4742528
15	562.0139298	187.3379766
15.1	673.6118714	224.5372905
15.2	692.3628823	230.7876274
15.3	0	0
15.4	0	0
15.5	48.62369105	16.20789702
15.6	53.45568043	17.81856014
15.7	54.35431383	18.11810461
15.8	55.26323383	18.42107794
15.9	56.652261	18.884087
16	59.03380371	19.67793457
16.1	62.96936934	20.98978978
16.2	69.07216329	23.02405443
16.3	77.97219635	25.99073212
16.4	90.17673167	30.05891056
16.5	105.7397766	35.2465922
16.6	123.698719	41.23290635
16.7	141.4712206	47.15707353
16.8	154.8339278	51.61130926
16.9	538.5855403	179.5285134
17	1727.899661	575.9665537
17.1	2407.40458	802.4681935
17.2	1143.348184	381.1160612
17.3	945.4757254	315.1585751
17.4	612.0174711	204.0058237
17.5	433.9950494	144.6650165
17.6	359.2912011	119.7637337
17.7	343.7917727	114.5972576
17.8	369.8509369	123.2836456
17.9	427.595501	142.5318337
18	513.3597262	171.1199087

Tension

Time	Tension (Output from MATLAB)	Tension / 3 (Used in SolidWorks)
18.1	617.5270439	205.842348
18.2	738.2509139	246.083638
18.3	210.6262658	70.20875528
18.4	0	0
18.5	13.38791672	4.462638905
18.6	52.63705654	17.54568551
18.7	53.93160556	17.97720185
18.8	54.77370487	18.25790162
18.9	55.85806887	18.61935629
19	57.67150811	19.22383604
19.1	60.75002852	20.25000951
19.2	65.68143967	21.89381322
19.3	73.09635225	24.36545075
19.4	83.59135598	27.86378533
19.5	97.50551617	32.50183872
19.6	114.4708904	38.15696346
19.7	132.7800462	44.26001541
19.8	148.9691598	49.65638659
19.9	331.8680383	110.6226794
20	934.5808016	311.5269339

LDPlot

Time	LDPlot (Output from MATLAB)	LDPlot * 100 (Used in SolidWorks)
0	-1.679707494	-335.9414987
0.1	1.448376563	289.6753126
0.2	2.32593578	465.1871559
0.3	2.021174108	404.2348215
0.4	1.837857301	367.5714602
0.5	1.726755043	345.3510085
0.6	1.656658962	331.3317925
0.7	1.605850236	321.1700472
0.8	1.555534112	311.1068224
0.9	1.489000186	297.8000372
1	1.394948168	278.9896336
1.1	1.272281603	254.4563205
1.2	1.131443373	226.2886746
1.3	0.989082174	197.8164347
1.4	0.859166022	171.8332043
1.5	2.06249442	412.498884
1.6	3.040215471	608.0430942
1.7	4.723760449	944.7520897
1.8	3.829570269	765.9140538
1.9	4.597833033	919.5666065
2	4.748723043	949.7446085
2.1	4.756754165	951.350833
2.2	4.88274292	976.5485839
2.3	5.064082113	1012.816423
2.4	5.203625908	1040.725182
2.5	5.265597592	1053.119518
2.6	5.281026996	1056.205399
2.7	5.264643425	1052.928685
2.8	5.229229846	1045.845969
2.9	4.643062792	928.6125583
3	-1.084629604	-216.9259208
3.1	-1.08166938	-216.333876
3.2	2.45016486	490.0329719
3.3	2.156221341	431.2442682
3.4	1.918125427	383.6250855
3.5	1.775786363	355.1572726
3.6	1.688254281	337.6508561
3.7	1.63036055	326.0721101
3.8	1.582278815	316.455763
3.9	1.526060628	305.2121255
4	1.447038779	289.4077558
4.1	1.338392686	267.6785372
4.2	1.20491703	240.983406
4.3	1.061153986	212.2307972
4.4	0.923529761	184.7059522
4.5	0.802233313	160.4466625

LDPlot

Time	LDPlot (Output from MATLAB)	LDPlot * 100 (Used in SolidWorks)
4.6	2.361350561	472.2701122
4.7	3.661823941	732.3647882
4.8	4.071730005	814.3460009
4.9	4.106689244	821.3378489
5	4.810493717	962.0987435
5.1	4.815870674	963.1741348
5.2	4.827640533	965.5281067
5.3	4.956824804	991.3649607
5.4	5.118808134	1023.761627
5.5	5.232214265	1046.442853
5.6	5.275690162	1055.138032
5.7	5.28241918	1056.483836
5.8	5.261223592	1052.244718
5.9	5.223723626	1044.744725
6	4.171309733	834.2619467
6.1	-1.356383276	-271.2766552
6.2	-0.93476498	-186.9529959
6.3	2.441882699	488.3765398
6.4	2.123550603	424.7101206
6.5	1.898485151	379.6970302
6.6	1.763736475	352.7472951
6.7	1.680879176	336.1758352
6.8	1.625820222	325.1640444
6.9	1.579130801	315.8261601
7	1.523041299	304.6082598
7.1	1.44315484	288.630968
7.2	1.333180451	266.6360902
7.3	1.198592708	239.7185416
7.4	1.054412756	210.8825513
7.5	0.917084413	183.4168825
7.6	0.796444058	159.2888116
7.7	2.457609062	491.5218125
7.8	3.83704687	767.409374
7.9	3.995345524	799.0691048
8	4.204544066	840.9088131
8.1	4.837718354	967.5436708
8.2	4.775635987	955.1271974
8.3	4.821857188	964.3714377
8.4	4.979273792	995.8547583
8.5	5.140975877	1028.195175
8.6	5.243611467	1048.722293
8.7	5.279080266	1055.816053
8.8	5.278112037	1055.622407
8.9	5.251149907	1050.229981
9	5.212054361	1042.410872

LDPlot

Time	LDPlot (Output from MATLAB)	LDPlot * 100 (Used in SolidWorks)
9.1	2.195922201	439.1844402
9.2	-1.695790518	-339.1581036
9.3	0.633290575	126.6581149
9.4	2.358801136	471.7602271
9.5	2.04279976	408.559952
9.6	1.850162186	370.0324372
9.7	1.733906374	346.7812749
9.8	1.661381542	332.2763084
9.9	1.610500085	322.100017
10	1.562391887	312.4783773
10.1	1.499960036	299.9920072
10.2	1.410779477	282.1558954
10.3	1.292030023	258.4060046
10.4	1.152712207	230.5424415
10.5	1.009255431	201.8510862
10.6	0.876662712	175.3325424
10.7	1.931609445	386.3218889
10.8	2.777915421	555.5830843
10.9	4.381141111	876.2282222
11	3.887024781	777.4049563
11.1	4.477375308	895.4750616
11.2	4.853005105	970.601021
11.3	4.803361951	960.6723901
11.4	4.869208812	973.8417624
11.5	5.021729589	1004.345918
11.6	5.176948326	1035.389665
11.7	5.25642683	1051.285366
11.8	5.282185661	1056.437132
11.9	5.274869138	1054.973828
12	5.24457098	1048.914196
12.1	5.206321789	1041.264358
12.2	0.827490046	165.4980091
12.3	-1.62317753	-324.635506
12.4	1.927421966	385.4843932
12.5	2.289973859	457.9947719
12.6	1.998284516	399.6569031
12.7	1.824083847	364.8167694
12.8	1.718427426	343.6854851
12.9	1.651673534	330.3347069
13	1.602698948	320.5397895
13.1	1.552926369	310.5852737
13.2	1.48583452	297.1669041
13.3	1.390524069	278.1048137
13.4	1.266538712	253.3077423
13.5	1.124932852	224.9865704

LDPlot

Time	LDPlot (Output from MATLAB)	LDPlot * 100 (Used in SolidWorks)
13.6	0.982586434	196.5172868
13.7	0.85324171	170.648342
13.8	2.056207141	411.2414283
13.9	3.038309574	607.6619148
14	4.728420202	945.6840403
14.1	3.847690789	769.5381578
14.2	4.609580624	921.9161248
14.3	4.756586248	951.3172497
14.4	4.758703823	951.7407645
14.5	4.878630351	975.7260702
14.6	5.059477662	1011.895532
14.7	5.200788911	1040.157782
14.8	5.267354705	1053.470941
14.9	5.280728909	1056.145782
15	5.263924602	1052.78492
15.1	5.226478398	1045.29568
15.2	4.970749222	994.1498444
15.3	-0.737988113	-147.5976226
15.4	-1.23605647	-247.211294
15.5	2.425753939	485.1507877
15.6	2.187415945	437.4831891
15.7	1.93670446	387.3408921
15.8	1.787100118	357.4200237
15.9	1.695340587	339.0681173
16	1.635272415	327.054483
16.1	1.586755866	317.3511732
16.2	1.531663321	306.3326641
16.3	1.45488889	290.977778
16.4	1.34873808	269.7476161
16.5	1.216958445	243.391689
16.6	1.073468711	214.6937422
16.7	0.934862103	186.9724206
16.8	0.812046221	162.4092443
16.9	2.369557471	473.9114943
17	3.661706769	732.3413538
17.1	4.052880237	810.5760473
17.2	4.077785598	815.5571195
17.3	4.793646898	958.7293797
17.4	4.739015711	947.8031423
17.5	4.793617561	958.7235121
17.6	4.957096489	991.4192978
17.7	5.128932112	1025.786422
17.8	5.238427087	1047.685417
17.9	5.276797868	1055.359574
18	5.277661465	1055.532293

LDPlot

Time	LDPlot (Output from MATLAB)	LDPlot * 100 (Used in SolidWorks)
18.1	5.252077352	1050.41547
18.2	5.213765306	1042.753061
18.3	2.204960503	440.9921006
18.4	-1.693451649	-338.6903299
18.5	0.635061899	127.0123798
18.6	2.363214216	472.6428431
18.7	2.046273536	409.2547072
18.8	1.852991967	370.5983934
18.9	1.736179749	347.2359498
19	1.662955118	332.5910235
19.1	1.611041537	322.2083074
19.2	1.561460793	312.2921586
19.3	1.4971936	299.43872
19.4	1.406162088	281.2324176
19.5	1.286075527	257.2151053
19.6	1.146339842	229.2679684
19.7	1.003372973	200.6745946
19.8	0.871773198	174.3546396
19.9	1.999691399	399.9382798
20	2.908190534	581.6381068

Power Conversion Mechanism

Time	OmegaG (Output form MATLAB)	Engaged / Disengaged	Angle of Gear (Input to SolidWorks)
0	48.12266473	Engaged	180
0.1	35.84264521	Engaged	135
0.2	24.70874603	Engaged	90
0.3	13.93686705	Engaged	45
0.4	3.31033865	Engaged	0
0.5	-7.30332478	Disengaged	45
0.6	-18.03331453	Disengaged	90
0.7	-28.99230852	Disengaged	135
0.8	-40.25639923	Disengaged	180
0.9	-51.83635135	Disengaged	225
1	-63.64038003	Disengaged	270
1.1	-75.43531335	Disengaged	315
1.2	-86.82626113	Disengaged	0
1.3	-97.29020385	Disengaged	45
1.4	-106.2935386	Disengaged	90
1.5	-111.3965094	Disengaged	135
1.6	-99.77336526	Disengaged	180
1.7	-37.33386194	Disengaged	225
1.8	38.72753839	Engaged	180
1.9	46.73966265	Engaged	135
2	52.90559979	Engaged	90
2.1	58.13314307	Engaged	45
2.2	59.54961354	Engaged	0
2.3	59.30667317	Engaged	315
2.4	58.56283008	Engaged	270
2.5	58.00230711	Engaged	225
2.6	57.68975636	Engaged	180
2.7	58.33373107	Engaged	135
2.8	59.98020212	Engaged	90
2.9	62.46027828	Engaged	45
3	54.53384319	Engaged	0
3.1	41.93218534	Engaged	315
3.2	30.23466572	Engaged	270
3.3	19.34175146	Engaged	225
3.4	8.665666921	Engaged	180
3.5	-1.930480943	Disengaged	225
3.6	-12.57889538	Disengaged	270
3.7	-23.40283544	Disengaged	315
3.8	-34.5001983	Disengaged	0
3.9	-45.91939374	Disengaged	45
4	-57.62613461	Disengaged	90
4.1	-69.46386121	Disengaged	135
4.2	-81.12031978	Disengaged	180
4.3	-92.12853874	Disengaged	225
4.4	-101.9399034	Disengaged	270
4.5	-110.07825	Disengaged	315

Power Conversion Mechanism

Time	OmegaG (Output form MATLAB)	Engaged / Disengaged	Angle of Gear (Input to SolidWorks)
4.6	-110.1534177	Disengaged	0
4.7	-84.9754532	Disengaged	45
4.8	2.004847142	Engaged	0
4.9	46.17057479	Engaged	315
5	46.43991648	Engaged	270
5.1	54.05654835	Engaged	225
5.2	58.36277913	Engaged	180
5.3	59.304733	Engaged	135
5.4	58.9565375	Engaged	90
5.5	58.2002521	Engaged	45
5.6	57.73712464	Engaged	0
5.7	57.57320049	Engaged	315
5.8	58.39633678	Engaged	270
5.9	60.20823474	Engaged	225
6	62.51512931	Engaged	180
6.1	53.24596605	Engaged	135
6.2	40.74455324	Engaged	90
6.3	29.20703801	Engaged	45
6.4	18.38123666	Engaged	0
6.5	7.759237214	Engaged	315
6.6	-2.793952071	Disengaged	0
6.7	-13.41104049	Disengaged	45
6.8	-24.21507001	Disengaged	90
6.9	-35.30269254	Disengaged	135
7	-46.7195692	Disengaged	180
7.1	-58.42659272	Disengaged	225
7.2	-70.259933	Disengaged	270
7.3	-81.8981061	Disengaged	315
7.4	-92.86531923	Disengaged	0
7.5	-102.6090826	Disengaged	45
7.6	-110.6584441	Disengaged	90
7.7	-109.1848115	Disengaged	135
7.8	-79.41785946	Disengaged	180
7.9	10.18580588	Engaged	135
8	46.52114834	Engaged	90
8.1	46.71319183	Engaged	45
8.2	55.57578745	Engaged	0
8.3	58.9760633	Engaged	315
8.4	59.44827096	Engaged	270
8.5	59.00798227	Engaged	225
8.6	58.23615628	Engaged	180
8.7	57.76274358	Engaged	135
8.8	57.81522562	Engaged	90
8.9	58.89551042	Engaged	45
9	60.93426539	Engaged	0

Power Conversion Mechanism

Time	OmegaG (Output form MATLAB)	Engaged / Disengaged	Angle of Gear (Input to SolidWorks)
9.1	61.45919995	Engaged	315
9.2	49.49294454	Engaged	270
9.3	37.15953459	Engaged	225
9.4	25.95949163	Engaged	180
9.5	15.19414881	Engaged	135
9.6	4.590504113	Engaged	90
9.7	-5.984406083	Disengaged	135
9.8	-16.66148744	Disengaged	180
9.9	-27.55748057	Disengaged	225
10	-38.75602376	Disengaged	270
10.1	-50.28022473	Disengaged	315
10.2	-62.05634252	Disengaged	0
10.3	-73.87401916	Disengaged	45
10.4	-85.36085766	Disengaged	90
10.5	-96.00522941	Disengaged	135
10.6	-105.2614037	Disengaged	180
10.7	-111.8652105	Disengaged	225
10.8	-104.7989617	Disengaged	270
10.9	-57.71154347	Disengaged	315
11	28.70743912	Engaged	270
11.1	46.42990446	Engaged	225
11.2	48.84438093	Engaged	180
11.3	56.29020546	Engaged	135
11.4	58.937144	Engaged	90
11.5	59.280595	Engaged	45
11.6	58.5718645	Engaged	0
11.7	57.95929022	Engaged	315
11.8	57.55497189	Engaged	270
11.9	57.84825335	Engaged	225
12	59.15961249	Engaged	180
12.1	61.38651639	Engaged	135
12.2	59.57788634	Engaged	90
12.3	46.94827328	Engaged	45
12.4	34.80115132	Engaged	0
12.5	23.76608919	Engaged	315
12.6	13.04809706	Engaged	270
12.7	2.465095014	Engaged	225
12.8	-8.115952364	Disengaged	270
12.9	-18.82431779	Disengaged	315
13	-29.77189469	Disengaged	0
13.1	-41.03272856	Disengaged	45
13.2	-52.61374203	Disengaged	90
13.3	-64.41701269	Disengaged	135
13.4	-76.20094353	Disengaged	180
13.5	-87.56134381	Disengaged	225

Power Conversion Mechanism

Time	OmegaG (Output form MATLAB)	Engaged / Disengaged	Angle of Gear (Input to SolidWorks)
13.6	-97.96865244	Disengaged	270
13.7	-106.8903355	Disengaged	315
13.8	-111.884048	Disengaged	0
13.9	-100.1299913	Disengaged	45
14	-37.70512776	Disengaged	90
14.1	38.53499543	Engaged	45
14.2	46.63284844	Engaged	0
14.3	52.89956655	Engaged	315
14.4	58.22072778	Engaged	270
14.5	59.66321579	Engaged	225
14.6	59.39291404	Engaged	180
14.7	58.65136881	Engaged	135
14.8	57.94863311	Engaged	90
14.9	57.77782298	Engaged	45
15	58.41473798	Engaged	0
15.1	60.06224611	Engaged	315
15.2	62.56271726	Engaged	270
15.3	55.79110796	Engaged	225
15.4	43.12050159	Engaged	180
15.5	31.29408714	Engaged	135
15.6	20.36155823	Engaged	90
15.7	9.662391865	Engaged	45
15.8	-0.944475198	Disengaged	90
15.9	-11.59130065	Disengaged	135
16	-22.40253622	Disengaged	180
16.1	-33.47896236	Disengaged	225
16.2	-44.87431703	Disengaged	270
16.3	-56.56287669	Disengaged	315
16.4	-68.40018093	Disengaged	0
16.5	-80.08829966	Disengaged	45
16.6	-91.17243188	Disengaged	90
16.7	-101.1065727	Disengaged	135
16.8	-109.4019155	Disengaged	180
16.9	-109.6769981	Disengaged	225
17	-84.55573913	Disengaged	270
17.1	2.790843883	Engaged	225
17.2	46.55471516	Engaged	180
17.3	46.77803004	Engaged	135
17.4	55.62709544	Engaged	90
17.5	59.03253968	Engaged	45
17.6	59.52719863	Engaged	0
17.7	59.02765589	Engaged	315
17.8	58.23045483	Engaged	270
17.9	57.77592362	Engaged	225
18	57.8130435	Engaged	180

Power Conversion Mechanism

Time	OmegaG (Output form MATLAB)	Engaged / Disengaged	Angle of Gear (Input to SolidWorks)
18.1	58.86426249	Engaged	135
18.2	60.87291679	Engaged	90
18.3	61.37746376	Engaged	45
18.4	49.39485964	Engaged	0
18.5	37.03443489	Engaged	315
18.6	25.80817049	Engaged	270
18.7	15.01507827	Engaged	225
18.8	4.38078655	Engaged	180
18.9	-6.227239189	Disengaged	225
19	-16.93902325	Disengaged	270
19.1	-27.8697018	Disengaged	315
19.2	-39.10022573	Disengaged	0
19.3	-50.64955352	Disengaged	45
19.4	-62.43813876	Disengaged	90
19.5	-74.24861808	Disengaged	135
19.6	-85.70220377	Disengaged	180
19.7	-96.28502426	Disengaged	225
19.8	-105.4570066	Disengaged	270
19.9	-111.4179159	Disengaged	315
20	-102.2647485	Disengaged	0